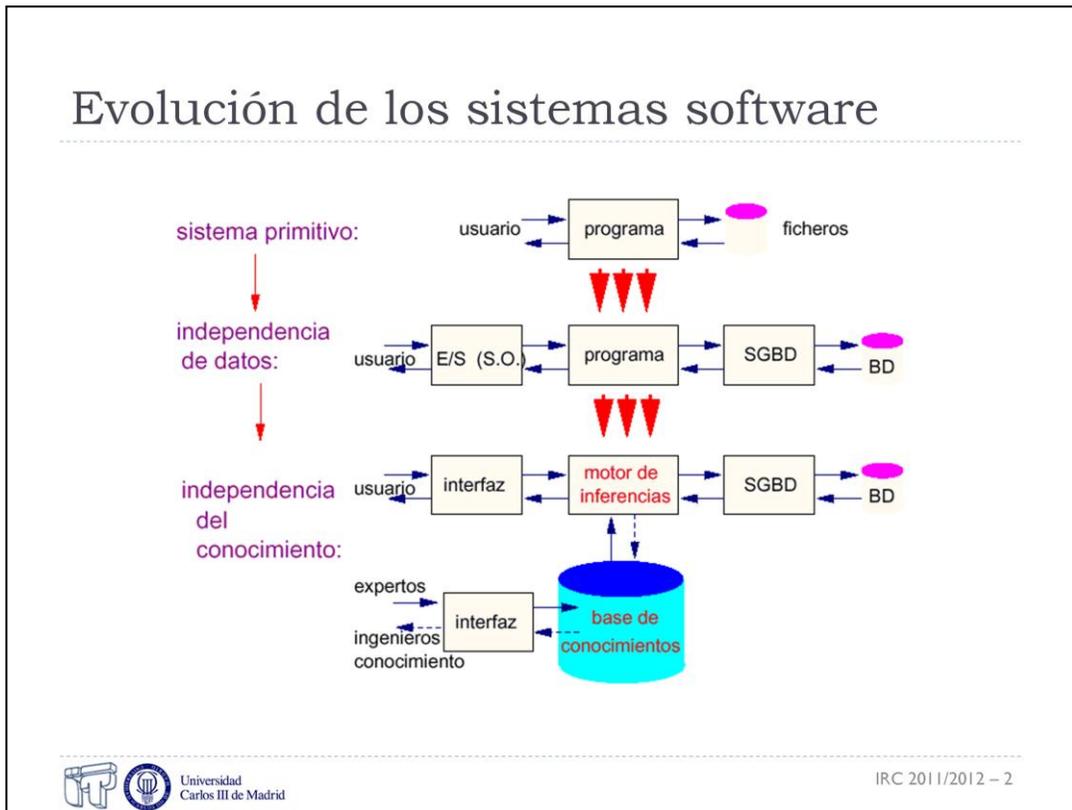


El objetivo de este Tema 3 es presentar los conceptos principales de los llamados “Sistemas Basados en Conocimiento”. Se empieza presentando las definiciones, razón de ser y fundamentos de este tipo de sistemas. A continuación se hace un repaso de diferentes modelos de representación del conocimiento existentes que se han utilizado en distintos sistemas, desde el más simple modelo de triplas (objeto, atributo, valor), marcos, redes semánticas, hasta modelos basados en diferentes lógicas. Más adelante se presenta una panorámica histórica de los sistemas basados en conocimiento y sus posibles aplicaciones, para por último terminar explicando las diferentes arquitecturas de diseño.



Los sistemas basados en conocimiento surgen como una evolución de los paradigmas de programación a lo largo de la historia de la informática. En los primeros tiempos de la computación, los programas, ante una petición del usuario, típicamente producían una salida accediendo a datos almacenados en ficheros con formatos particulares de cada aplicación.

Más adelante surgieron las bases de datos, que abstraían la capa de almacenamiento/persistencia de datos, eliminando esos ficheros, facilitando así la gestión de la información necesaria para la aplicación. Además aparecen los conceptos de la interfaz gráfica del sistema operativo, facilitando la entrada/salida.

Los sistemas basados en conocimiento son una nueva evolución: en estos sistemas, los datos, la “base de conocimientos”, se convierten en lo más importante, y el código que los explota, el “motor de inferencias”, se generaliza según diferentes paradigmas. Esta base de conocimientos la produce un usuario experto en el dominio, los ingenieros del conocimiento, que son diferentes a los programadores de la interfaz o del propio motor de inferencias.

Sistemas tradicionales vs. SBC

▶ Sistemas tradicionales

- ▶ El usuario representa el conocimiento y además debe definir cómo interacciona con el conocimiento previo y debe restablecer la secuencia de instrucciones de procesamiento

▶ Sistemas Basados en Conocimiento

- ▶ El usuario (experto) define el conocimiento y el sistema lo integra directamente en la aplicación existente

Esta diapositiva presenta la diferencia fundamental entre los sistemas tradicionales y los sistemas basados en conocimiento.

Ingeniería del conocimiento

- ▶ La **ingeniería del conocimiento** surge en los años 70 y se enfoca al diseño y construcción de sistemas basados en conocimiento, diferenciándose de la ingeniería del software en:

- ▶ Conocimiento humano incompleto, inconsistente, impreciso, incierto, tolerante
- ▶ Razonamiento y respuestas del sistema inseguros
- ▶ Conocimiento evolutivo, posibilidad de aprendizaje
- ▶ No aplicable el ciclo de vida en cascada
- ▶ Figura del “experto” en los proyectos de desarrollo
 - ▶ Aquel que tiene el conocimiento necesario para resolver problemas en un determinado campo o dominio de conocimiento

La ingeniería del conocimiento surge como una variante de la ingeniería del software tradicional en cuanto a que se enfoca al diseño y construcción de sistemas basados en conocimiento.

Dos paradigmas de sistemas inteligentes

▶ **SBC: orientación estructural (cognoscitiva)**

- ▶ poder resolver determinados problemas radica en un “saber”, o conocimiento específico, sobre el tipo de problemas, más que en capacidades intelectivas generales
- ▶ separación entre el conocimiento codificado en la máquina (base de conocimientos) y los mecanismos deductivos (motor de inferencias)

▶ **Sistemas expertos: orientación funcional (conductista)**

- ▶ se analiza la actividad de un experto humano cuando resuelve problemas en un área muy concreta y se intenta emularlo o ayudarlo
- ▶ capacidad para adquirir incrementalmente experiencia
- ▶ y capacidad para conversar con los usuarios y explicarles sus líneas de razonamiento

Históricamente se ha utilizado también el nombre de “sistemas expertos”, con una distinción sutil. Al hablar de “sistemas basados en conocimiento” en general, simplemente se está haciendo referencia a que lo más importante del sistema es el conocimiento que almacena y que gestiona, es decir, una visión estructural. Sin embargo, el término “sistema experto” se refiere a un sistema que imita la actividad de un experto humano para resolver una determinada tarea en un dominio específico, es decir, una visión funcional del sistema.



Así, un sistema experto es un tipo particular de sistema basado en conocimiento, el que imita al experto humano. Hay sistemas basados en conocimiento que no son sistemas expertos, por ejemplo, una enciclopedia electrónica. Un sistema basado en conocimiento es un tipo de sistema inteligente, en el sentido de almacenar/gestionar/estar relacionado con el saber o capacidades humanas.

Índice

- ▶ **Introducción**
- ▶ **Representación del conocimiento**
 - ▶ Modelo del procesamiento de la información
 - ▶ Lenguajes de representación del conocimiento
 - ▶ Triplas OAV
 - ▶ Redes semánticas
 - ▶ Marcos
 - ▶ Lógica de predicados (de primer orden)
- ▶ **Historia y aplicaciones de los SBC**
- ▶ **Arquitectura de los SBC**

Índice del resto del tema.

Representación del conocimiento

Cómo se puede representar explícitamente el “conocimiento” que un experto utiliza para resolver problemas, para su almacenamiento y empleo en un sistema software de forma versátil y eficiente

Cómo representar el conocimiento almacenado en el “cerebro” de la máquina (memoria)

En este punto se va a estudiar diferentes formas de representar conocimiento en la memoria de un ordenador para optimizar su utilización de forma versátil y rápida.

Modelos de representación

Modelos derivados de la psicología

▶ **subjetivismo**

- ▶ von Helmholtz, James (1880)

▶ **conductismo**

- ▶ Watson (1878-1958) → *pequeño Albert*
- ▶ Thorndike (1874-1949) → *ley del efecto (gatos en laberinto)*
- ▶ Paulov (1849-1936) → *condicionamiento clásico (perro de Paulov)*
- ▶ Skinner (1904-1990) → *condicionamiento operante/instrumental (palomas y comida)*

▶ **cognitivismo/psicología cognitiva**

- ▶ Jean Piaget (1896-1980) → *lógica, asimilación, acomodación*
- ▶ Lev Vigotsky (1896-1934) → *psicología del aprendizaje*
- ▶ George Miller → *racimos (1956)*
- ▶ Ross Quillian → *memoria semántica (1968)*
- ▶ Allen Newell y Herbert Simon → *modelo de P.I. (1972)*

Los modelos de representación del conocimiento se inspiran en modelos derivados de la psicología. En los primeros tiempos del subjetivismo, los seres vivos se consideraban tan complicados que ni siquiera se planteaba su estudio. Sin embargo, a partir de los trabajos de Watson, Thorndike, Paulov, Skinner y otros conductistas, se empieza a estudiar el ser desde el punto de vista de estímulo/respuesta: aunque el ser todavía es una caja negra cuyo funcionamiento interno se desconoce, se empieza a modelar que ante un estímulo determinado se producen una serie de respuestas predecibles, observables y modificables de alguna forma determinada. A partir de Jean Piaget, el padre de la psicología moderna, se desarrolla el cognitivismo, en el que ya se aborda el estudio del razonamiento de los seres vivos como cajas blancas, interesándose en los procesos internos de razonamiento, pensamiento, memoria, aprendizaje, etc.

Constructivismo

- ▶ **El conocimiento no se adquiere desde fuentes exteriores, sino que cada individuo debe **crear**lo en su mente**
 - ▶ El conocimiento es activamente construido por el sujeto cognoscente, no pasivamente recibido del entorno
 - ▶ Llegar a conocer es un proceso adaptativo que organiza el mundo experiencial de la persona
- ▶ **Características generales:**
 - ▶ Todo conocimiento es construido
 - ▶ Existen estructuras cognitivas que activan los procesos de construcción
 - ▶ Las estructuras cognitivas están en desarrollo continuo, la actividad transforma las estructuras existentes

Conceptos y características del constructivismo, sobre la creación del conocimiento (aprendizaje).

Teoría del desarrollo (Piaget)

- ▶ No es posible dar información a un ser humano y esperar que inmediatamente la entienda y la use. Por el contrario, la persona debe **“construir” su propio conocimiento** (modelos mentales), fundamentalmente a través de la experiencia (**desarrollo**)
- ▶ Los modelos mentales se desarrollan y se hacen más sofisticados con nuevas experiencias a través de la **asimilación** y la **acomodación**
- ▶ El papel del profesor es proveer un entorno rico para que el aprendiz lo explore espontáneamente
 - ▶ Un entorno lleno de elementos interesantes y experiencias que favorezcan la asimilación y la acomodación motiva al aprendiz
 - ▶ El aprendiz debería tener libertad para entender y construir significado a su propio ritmo, a través de experiencias personales
- ▶ **En resumen:**
 - ▶ El aprendizaje es un proceso activo, en el que cometer errores, tener que volver atrás y encontrar soluciones es vital, de cara a restablecer el equilibrio mediante asimilación y acomodación
 - ▶ El aprendizaje debería ser “auténtico” y “real”, planteando al aprendiz problemas significativos (y no meros ejercicios) y actividades realistas

La teoría del desarrollo de Jean Piaget, base de gran parte de la pedagogía moderna.

Aprendizaje social (Vygotsky)

- ▶ Una componente fundamental del aprendizaje es el valor añadido que aporta la **interacción social**, no sólo alumno-profesor, sino dentro del grupo de alumnos (interacción alumno-grupo)
- ▶ El origen de las facultades que posibilitan el aprendizaje está en la relación entre personas
- ▶ El nivel de desarrollo más elevado (*Zone of Proximal Development, ZPD*), sólo se consigue mediante la interacción en el grupo:
 - ▶ la capacidad que se puede obtener en un medio social, bien basado en la función orientativa del profesor o tutor o en el aprendizaje compartido de trabajos en equipo, etc. es mucho mayor de la que se conseguiría con el aprendizaje en solitario
- ▶ Cada estudiante puede ser constructor de su propio aprendizaje si cuenta con la ayuda de un mediador competente responsable que:
 - ▶ Asegure que la transmisión de información se haga de forma correcta (sin errores)
 - ▶ Ubique al alumno en un entorno de interacción social completa mediante trabajos en equipo, debates y otras actividades que permitan el aprendizaje compartido

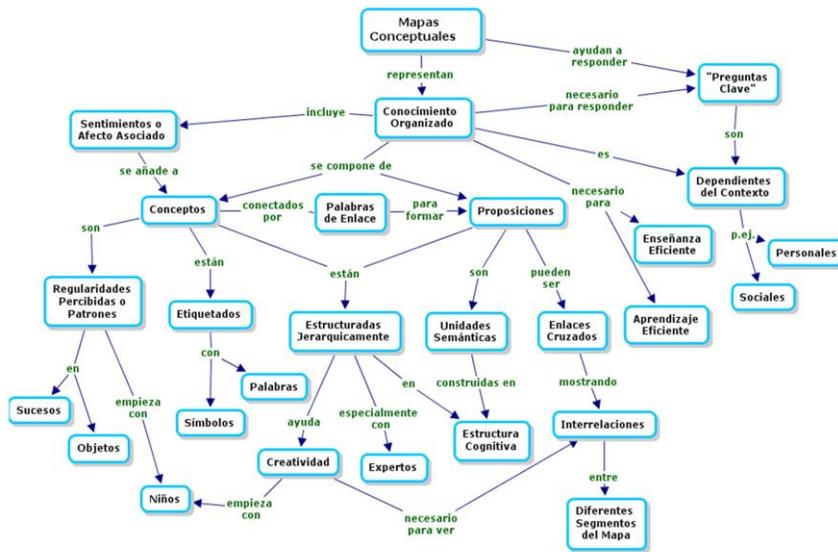
El aprendizaje social de Vygotski hace énfasis en que los seres vivos son seres sociales, y que también se aprende del entorno social. Por ejemplo, un jugador de tenis mejora su juego al enfrentarse a jugadores de nivel parecido o ligeramente superior al suyo.

Teoría del Aprendizaje Receptivo Significativo (Ausubel)

- ▶ Los contenidos son expuestos ante el alumno, no descubiertos por éste y sin limitarse a memorizar la información
- ▶ Mecanismo por excelencia en el ser humano (sobre todo adulto) para adquirir y almacenar ideas e informaciones
- ▶ La teoría de Ausubel trata de cómo los materiales (orales o textuales) son aprendidos en el contexto de una clase o autoestudio
- ▶ Aprender nuevos materiales depende en gran medida de las estructuras cognitivas preexistentes (lo que la persona ya sabe). Así, la información nueva será más significativa y fácilmente aprendida si está relacionada con los conocimientos previos del estudiante
- ▶ **Ideas:**
 - ▶ Las ideas más generales deberían ser presentadas al comienzo de la lección
 - ▶ La instrucción debería integrar los nuevos materiales con información que haya sido previamente presentada o aprendida
 - ▶ Los instructores deberían usar multitud de ejemplos, concentrándose en las similitudes y en las diferencias
 - ▶ Lo que el estudiante ya sabe es el factor más importante de cara al aprendizaje posterior
- ▶ Base de los Mapas Conceptuales

La teoría del Aprendizaje Receptivo Significativo de Ausubel trata de cómo los materiales de estudio llegan al alumno en el contexto del aprendizaje y son la base de los mapas conceptuales.

Mapas conceptuales (Novak)



Los mapas conceptuales de Novak son una potente herramienta para facilitar la comprensión de un problema y la adquisición de conocimiento sobre un dominio determinado.

Triángulo de Tennyson

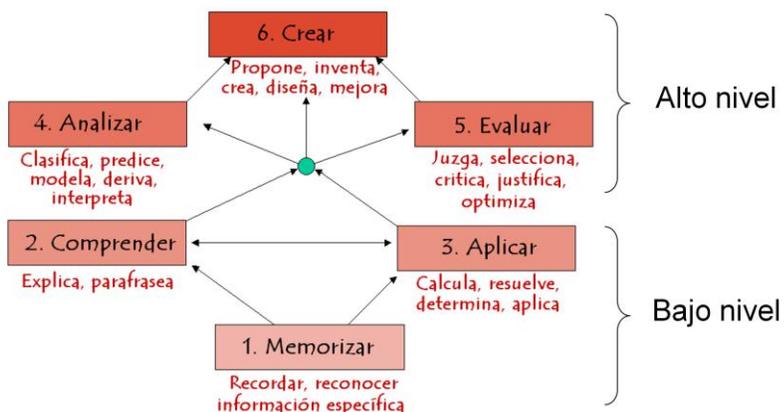
- ▶ Tres estrategias de pensamiento (complejidad cognitiva creciente):
 - ▶ Estrategias de **recuerdo** (memoria)
 - ▶ Actividades para el aprendizaje de los conceptos básicos, que deberán entenderse y almacenarse en la memoria como información a la que recurrir después
 - ▶ Estrategias de **resolución de problemas**
 - ▶ Conocimiento contextual que se proporcionará mediante actividades prácticas, en las que los alumnos puedan empezar a aplicar los conocimientos teóricos en la resolución de pequeños problemas prácticos
 - ▶ Esto les hace ir acumulando estrategias para el empleo real de los conocimientos teóricos, a la vez que produce un mejor entendimiento y asimilación de los mismos
 - ▶ Estrategias de **creatividad**
 - ▶ La creación de nuevo conocimiento vendrá a través de la realización de un proyecto global, que abarque gran parte del tiempo de impartición del curso y que suponga el planteamiento de un problema complejo para cuya resolución deberán aplicarse de forma creativa (esto es, ligeramente diferente a lo visto en la teoría y en las actividades prácticas) una gran variedad de conceptos y habilidades de los adquiridos en el curso
 - ▶ Esta actividad supondrá para los estudiantes la personalización de los conocimientos adquiridos, desarrollando metodologías y aproximaciones propias a su empleo, así como la interiorización definitiva de los mismos

Tennyson expone que las estrategias de pensamiento son de complejidad cognitiva creciente y las divide en tres posibles actividades de alto nivel: recordar, resolver problemas y crear.

La Taxonomía de Bloom

Taxonomía de Objetivos del Proceso de Aprendizaje: lo que el estudiante debe haber adquirido tras un proceso de aprendizaje

(Benjamin Bloom, Univ. Chicago, EEUU)

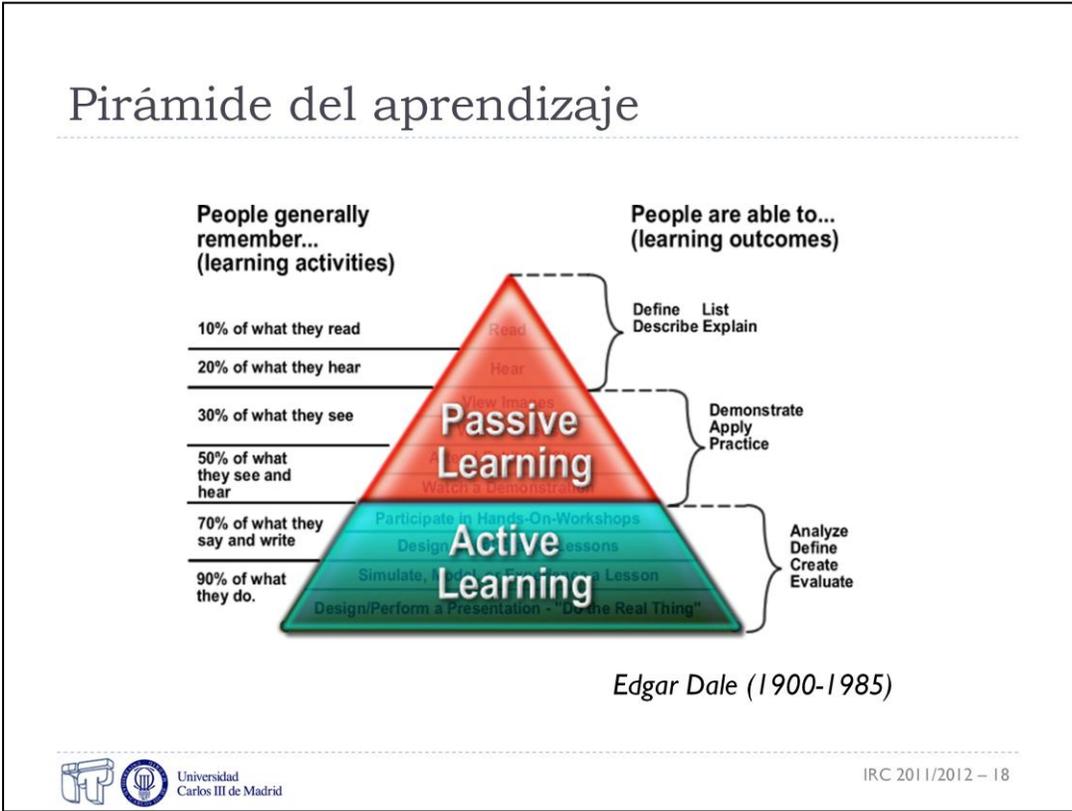


Los Objetivos del Proceso de Aprendizaje, de Benjamin Bloom, indican lo que se espera que haya aprendido el estudiante tras un proceso de aprendizaje. Se han utilizado para la definición del modelo del conocimiento del Espacio Europeo de Educación Superior (EEES).

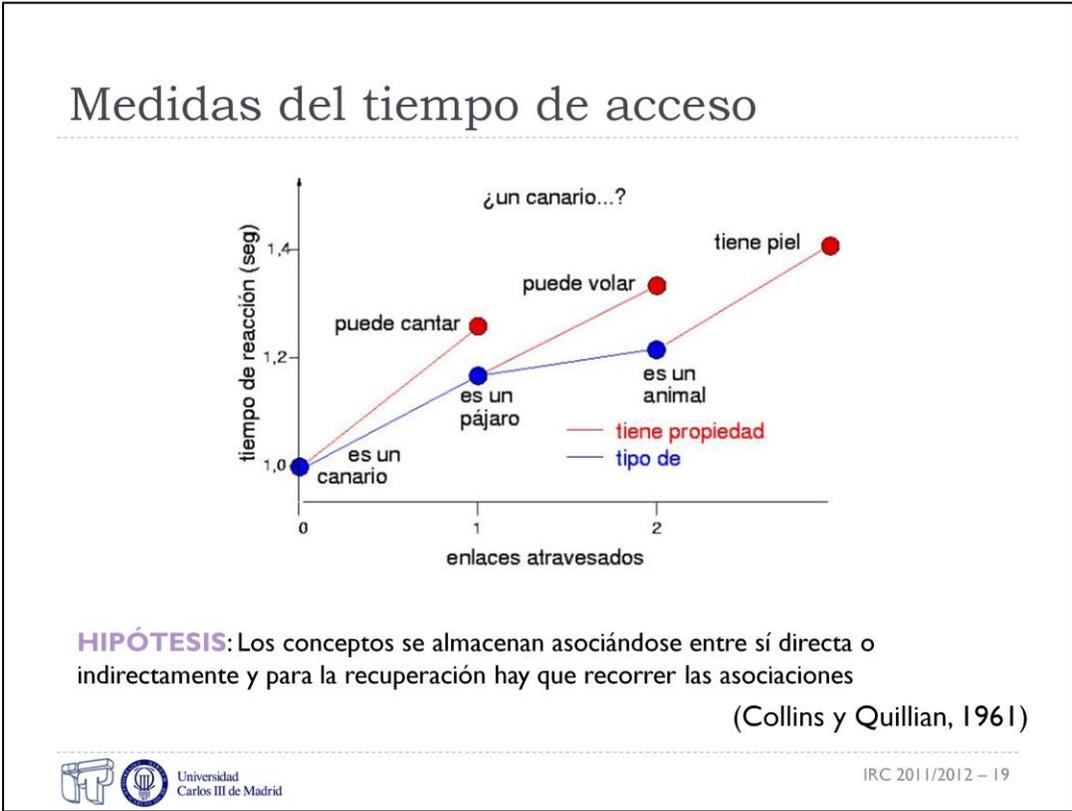
La Taxonomía de Bloom (2)

Memorizar	Recordar, reconocer información específica como hechos, sucesos, fechas, nombres, símbolos, teorías, definiciones y otros.
Comprender	Entender el material que se ha aprendido. Esto se demuestra cuando se presenta la información de otra forma, se transforma, se buscan relaciones, se asocia, se interpreta (explica o resume) o se presentan posibles efectos o consecuencias.
Aplicar	Usar el conocimiento y destrezas adquiridas en nuevas situaciones.
Analizar	Descomponer el todo en sus partes, se solucionan problemas a la luz del conocimiento adquirido y razonar.
Evaluar	Enjuiciar (selecciona, critica, justifica) sobre la base de criterios establecidos.
Crear	Crear, mejorar, optimizar o hacer algo original.

Niveles de Objetivos del Proceso de Aprendizaje de Bloom.



Pirámide del aprendizaje de Edgar Dale, que refleja la capacidad de producir adquisición efectiva de conocimiento a partir de tareas específicas.



Los estudios de Collins y Quilian, a partir de medidas del tiempo medio de respuesta a diferentes preguntas, establecen que en el cerebro de los seres vivos, los conceptos se almacenan asociándose entre sí.

Tipos de memoria

▶ Memoria episódica

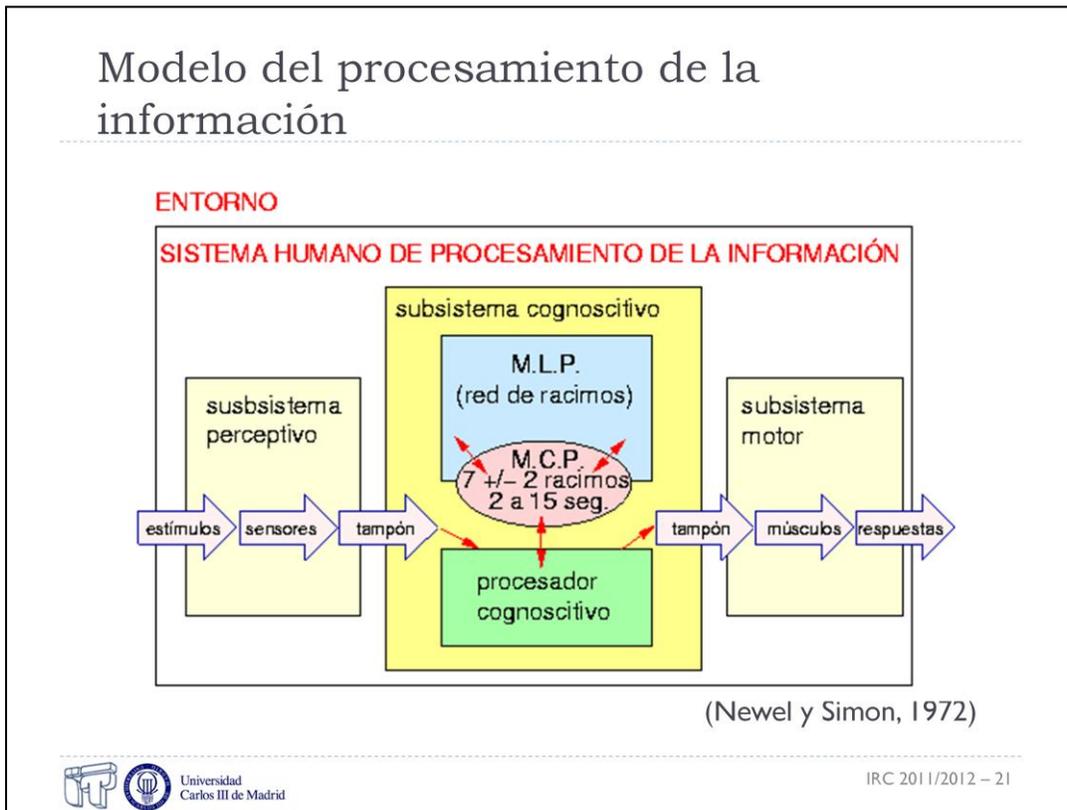
- ▶ recuerdo de eventos experimentados (episodios)
- ▶ surge tras una vivencia

▶ Memoria semántica

- ▶ almacenamiento de información variada: vocabulario, hechos, relaciones, procedimientos... sin referencia a cómo, dónde o cuándo se han adquirido esos conocimientos
- ▶ surge tras un proceso de abstracción

(Ross Quillian, 1968)

Dos tipos de memoria: episódica (activo, fruto de las experiencias) y semántica (conocimientos generales).



Propuesta de Newell y Simon que modela el sistema nervioso central de los seres humanos como un sistema computacional de procesamiento de la información.

Tipos de conocimiento

► Declarativo

```
superior(X,Y) :- jefe(X,Y).  
superior(X,Y) :- jefe(X,Z), superior(Z,Y).
```

► Procedimental

```
boolean superior(individuo X, individuo Y, conjunto C) {  
    if (jefe(X,Y)) return true; else {  
        C = sacar(C, X);  
        C = sacar(C, Y);  
        while (!vacio(C)) {  
            Z = buscar_en(C);  
            if (jefe(X,Z))  
                if (superior(Z,Y)) return true;  
            C = sacar(C, Z);  
        }  
        return false;  
    }  
}
```

(Ross Quillian, 1968)

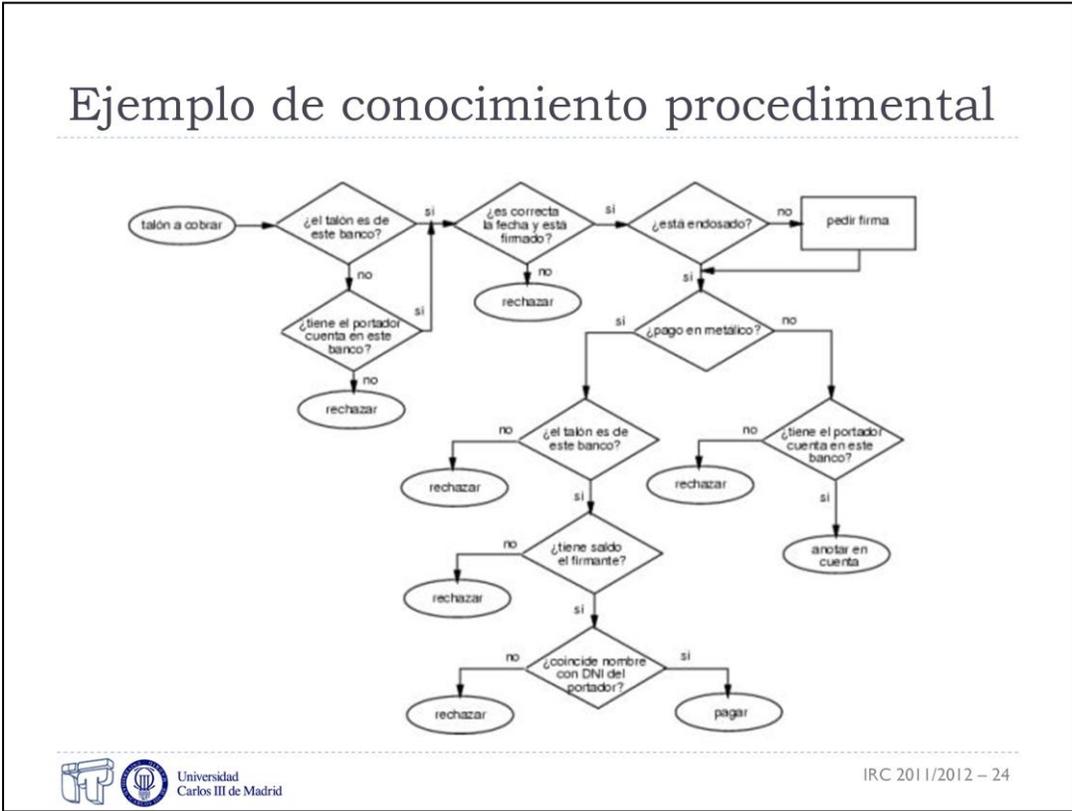
Quillian propone dos tipos de conocimiento: declarativo, donde se exponen los hechos y las reglas que rigen un determinado dominio, y procedimental, en el que se dan instrucciones paso a paso sobre cómo realizar una tarea.

Ejemplo de conocimiento declarativo

- (1) SI NO talón_de_este_banco
Y NO portador_tiene_cuenta
ENTONCES rechazar
- (2) SI talón_de_este_banco
O portador_tiene_cuenta
ENTONCES talón_aceptable
- (3) SI fecha_correcta
Y talón_firmado
ENTONCES talón_cumplimentado
- (4) SI NO talón_cumplimentado
ENTONCES rechazar
- (5) SI talón_cumplimentado
Y NO talón_endosado
ENTONCES pedir_firma
Y talón_endosado
- (6) SI talón_aceptable
Y talón_cumplimentado
Y talón_endosado
ENTONCES talón_completo
- (7) SI pago_metalico
Y NO talón_de_este_banco
ENTONCES rechazar
- (8) SI pago_metalico
Y talón_de_este_banco
Y NO firmante_saldo
ENTONCES rechazar
- (9) SI pago_metalico
Y NO portador_DNI
ENTONCES rechazar
- (10) SI NO pago_metalico
Y NO portador_tiene_cuenta
ENTONCES rechazar
- (11) SI talón_completo
Y NO pago_metalico
Y portador_tiene_cuenta
ENTONCES anotar
- (12) SI talón_completo
Y pago_metalico
Y talón_de_este_banco
Y firmante_saldo
Y portador_DNI
ENTONCES pagar
- (13) SI (hoy - fecha_talón) > 0
Y (hoy - fecha_talón) < 90
ENTONCES fecha_correcta



Ejemplo de conocimiento declarativo: normativa de pagos de talones bancarios.



El mismo ejemplo del talón bancario pero con conocimiento procedimental. En este caso, mucho más sencillo de entender en esta forma.

Programación funcional/imperativa

Quicksort in Haskell

```
qsort [] = []
qsort (x:xs) = qsort elts_lt_x ++ [x] ++ qsort elts_greq_x
  where
    elts_lt_x = [y | y <- xs, y < x]
    elts_greq_x = [y | y <- xs, y >= x]
```

Quicksort in C

```
qsort( a, lo, hi ) int a[], hi, lo;
{
  int h, l, p, t;

  if (lo < hi) {
    l = lo;
    h = hi;
    p = a[hi];

    do {
      while ((l < h) && (a[l] <= p))
        l = l+1;
      while ((h > l) && (a[h] >= p))
        h = h-1;
      if (l < h) {
        t = a[l];
        a[l] = a[h];
        a[h] = t;
      }
    } while (l < h);

    t = a[l];
    a[l] = a[hi];
    a[hi] = t;

    qsort( a, lo, l-1 );
    qsort( a, l+1, hi );
  }
}
```



Ejemplo del algoritmo de quicksort usando conocimiento imperativo (con un lenguaje de programación imperativo, C, a la derecha) y conocimiento declarativo (con un lenguaje de programación funcional, Haskell, a la izquierda). En este caso, mucho más sencillo expresarlo de forma declarativa.

Lenguajes de representación

▶ **Requisitos:**

- ▶ **sintaxis formalizada**, para poder diseñar adecuadamente un procesador (“motor de inferencias”)
 - ▶ **semántica bien definida** y que permita la implementación en el procesador de algoritmos de razonamiento eficientes
 - ▶ desde el **punto de vista pragmático**, expresividad para representar el conocimiento lo menos forzosamente posible
- ▶ Cada lenguaje suele estar soportado por una herramienta

Requisitos de los lenguajes de representación del conocimiento.

Lenguajes de representación

- ▶ Triplas Objeto-Atributo-Valor
- ▶ Redes semánticas
- ▶ Marcos
- ▶ Lógica de predicados

Se van a estudiar estos cuatro modelos de lenguajes de representación.

Triplas objeto-atributo-valor

(objeto, atributo, valor)

<i>objetos</i>	<i>atributos</i>	<i>valores</i>
talón	banco	este_banco, otro_banco
	aceptable	sí, no
	cumplimentado	sí, no
	endosado	sí, no
	completo	sí, no
	fecha_correcta	sí, no
	fecha	M/D/A
portador	acción	pagar, rechazar, anotar
	tiene_cuenta	sí, no
firmante	DNI	sí, no
	saldo	sí, no
pago	modo	metalico, en_cuenta

```
IF (talon, banco, otro_banco)
  AND (portador, tiene_cuenta, no)
THEN (talon, accion, rechazar)
.....

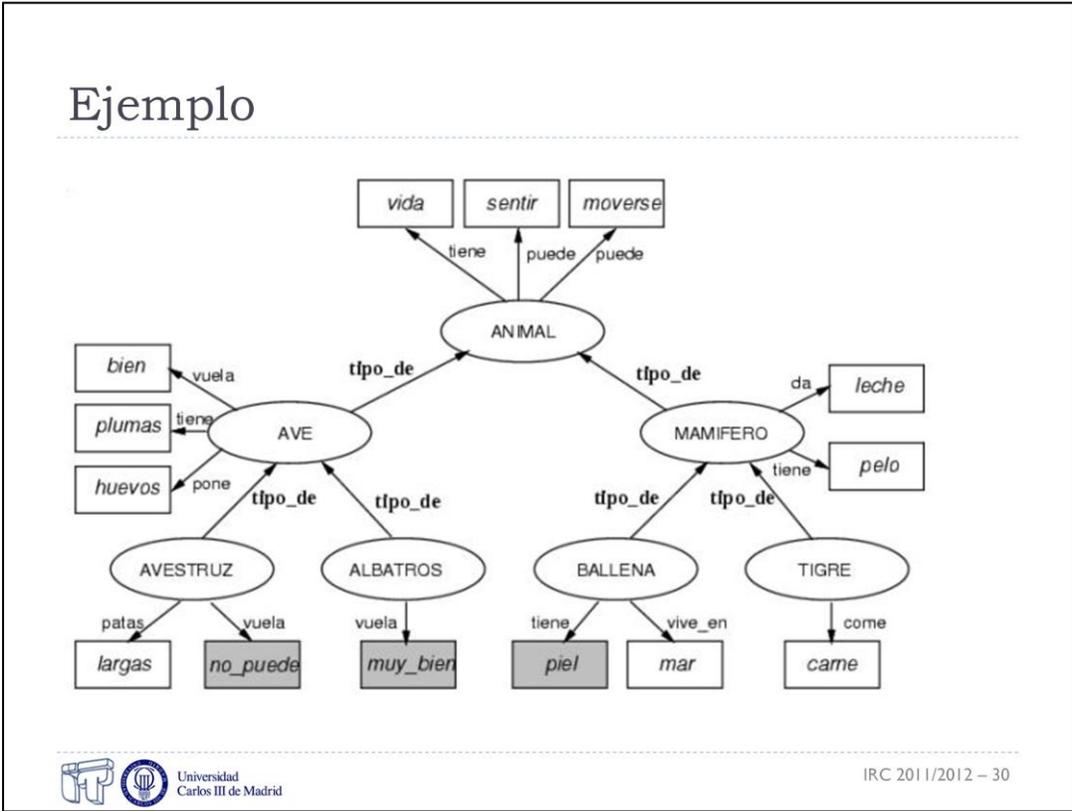
IF (talon, fecha, X)
  AND ($system, $date, Y)
  AND ((Y-X) > 0)
  AND ((Y-X) < 90)
THEN (talon, fecha_correcta, si)
```

Las triplas (objeto, atributo, valor) son el modelo más básico. Han evolucionado con el tiempo en los atributos de una clase en el paradigma de la programación orientada a objetos.

Redes semánticas

- ▶ Lenguaje gráfico fácilmente traducible a forma simbólica, muy útil para representar conocimiento **taxonómico** (aquél que permite agrupar los elementos del universo en una jerarquía de clases y subclases entre las cuales existe una relación de herencia)
- ▶ Permite una **representación estructurada del conocimiento factual** y de ciertos tipos de **conocimiento normativo**, y de expresar de manera muy fácil las **excepciones** en los valores de propiedades heredadas

Las redes semánticas es un lenguaje gráfico muy sencillo y útil para representar conocimiento taxonómico.



Ejemplo de red semántica en el dominio de la zoología. Las elipses son conceptos, los rectángulos son propiedades, y se han marcado con negrita aquellas propiedades que sobrescriben propiedades heredadas.

Elementos de las redes semánticas

▶ Nodos

- ▶ de objetos
- ▶ de clases
- ▶ de valores de propiedades o atributos

▶ Arcos orientados

- ▶ relaciones binarias entre objetos y clases (entre ellas, las que permiten la herencia de propiedades: *es_un* y *tipo_de*)
- ▶ relaciones binarias entre objetos o clases y valores de propiedades (“*tiene*”, “*vuela*”, “*pone*”...)

Elementos de las redes semánticas.

Marcos (*frames*)

- ▶ Forma de expresar las redes semánticas textualmente pero además pueden incluir representaciones de conocimiento procedimental
- ▶ Cada nodo correspondiente a un objeto o a una clase se convierte en un marco, que consta de una primera línea con el nombre del marco y una sucesión de líneas, llamadas “*ranuras*” (*slots*)

Los marcos o “frames” son otro modelo de representación del conocimiento.

Ejemplo de marco

```
marco_avestruz
  descripcion: caracterización básica de un avestruz
  autor      : Prof. Ornito
  fecha     : 31/12/2000
  tipo_de   : ave
  patas     : largas
  vuela     : (excl) no_puede
  detalles  : if (necesario) {buscar en enciclopedia}
```

Ejemplo de marco.

Lógica de predicados (de primer orden)

La lógica de predicados de primer orden estudia
razonamientos de tipo deductivo, exactos (verdadero/falso)
y basados en predicados de primer orden

```
hijo(X,Y) :- padre(Y,X), hombre(X).  
hijo(X,Y) :- madre(Y,X), hombre(X).  
hija(X,Y) :- padre(Y,X), mujer(X).  
hija(X,Y) :- madre(Y,X), mujer(X).  
abuelo(X,Y) :- padre(X,Z), padre(Z,Y), hombre(X).  
abuela(X,Y) ...
```

El modelo de representación del conocimiento de la lógica de primer orden. [Los razonamientos lógicos se estudian en profundidad en el Tema 4 y 5].

Ventajas e inconvenientes

▶ Ventajas

- ▶ Modelo (funcional) para los razonamientos humanos
- ▶ Alto grado de formalización (sintaxis y semántica)
- ▶ Separación conocimiento (reglas)/razonamiento (inferencia)
- ▶ Otros esquemas se basan o pueden expresarse con ella
- ▶ Suficientemente expresiva para muchos dominios

▶ Inconvenientes

- ▶ A veces es demasiado expresiva
 - ▶ los algoritmos de inferencia para el caso general son complejos y hay ciertos tipos de problemas en los que bastaría un lenguaje con algoritmos más sencillos
- ▶ A veces se queda corta
 - ▶ problemas para razonar con conocimiento incierto, impreciso y subjetivo

Ventajas e inconvenientes de la lógica de primer orden.

Ejemplo (red semántica a lógica)

propiedades

```
prop(animal, tiene, vida).  
prop(animal, puede, sentir).  
prop(animal, puede, moverse).  
prop(ave, vuela, bien).  
prop(ave, tiene, plumas).  
prop(ave, pone, huevos).  
prop(mamifero, da, leche).  
prop(mamifero, tiene, pelo).  
prop(avestruz, patas, largas).  
prop(ballena, vive, mar).  
prop(tigre, come, carne).
```

excepciones

```
excep(avestruz, vuela, nopuede).  
excep(albatros, vuela, muybien).  
excep(ballena, tiene, piel).
```

Ejemplo de representación de una red semántica con lógica de predicados de primer orden, en concreto, usando el lenguaje Prolog. Se modelan las propiedades de cada objeto y las excepciones de herencia de las propiedades.

Ejemplo (red semántica a lógica) (2)

relaciones

```
tipo_de(ave, animal).  
tipo_de(mamifero, animal).  
tipo_de(avestruz, ave).  
tipo_de(albatros, ave).  
tipo_de(ballena, mamifero).  
tipo_de(tigre, mamifero).
```

herencia

```
hereda_de(X,Y) :- tipo_de(X,Y).  
hereda_de(X,Y) :- tipo_de(X,Z), hereda_de(Z,Y).  
  
prop(Clase, Atributo, Valor) :- hereda_de(Clase, ClasePadre),  
    prop(ClasePadre, Atributo, Valor), not excep(Clase, Atributo,  
    Valor).  
excep(Clase, Atributo, Valor) :- excep(Clase, Atributo, Valor).
```

Continuación del ejemplo, presentando las relaciones y las reglas de herencia.

Ejemplo (red semántica a lógica) (3)

```
?- prop(avestruz, tiene, pelo).  
NO
```

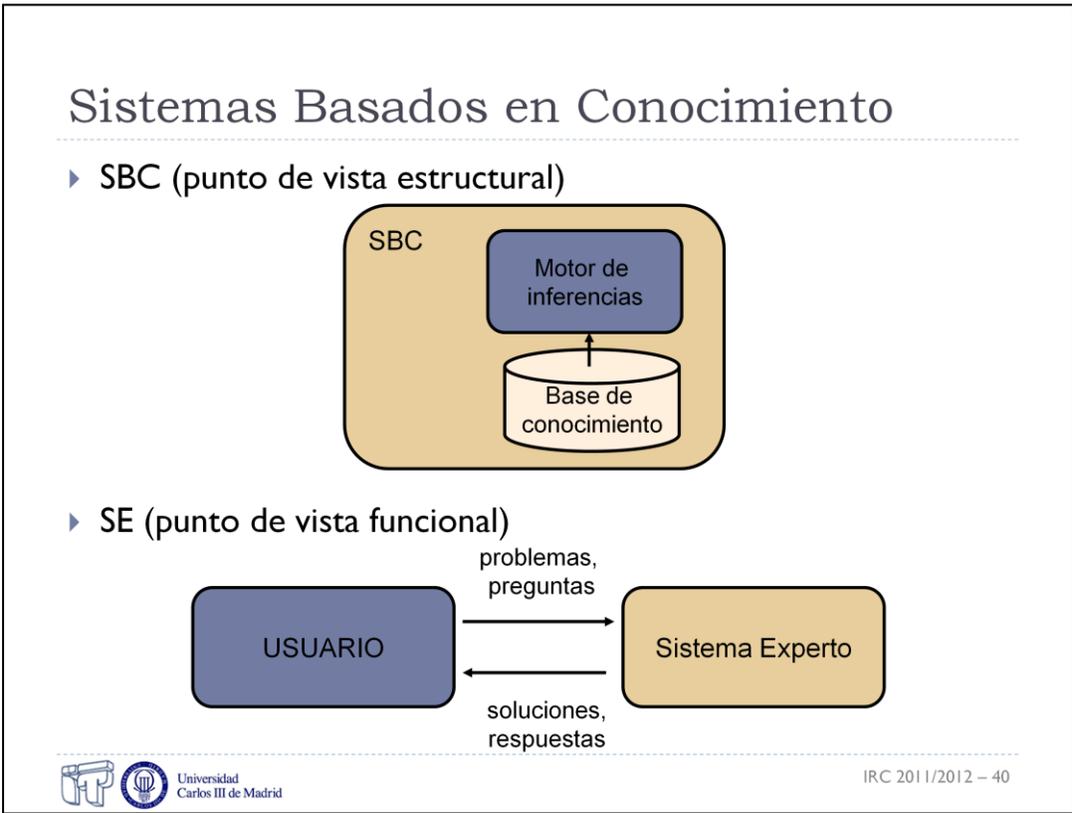
```
?- prop(Quien, vuela, Como).  
Quien=ave Valor=bien.  
Quien=avestruz Valor=no_puede.  
Quien=albatros Valor=muy_bien.
```

Consultas en una consola Prolog para explotar el conocimiento de la red semántica. El avestruz NO tiene pelo, y una lista de quién vuela y cómo.

Ontología

- ▶ En **filosofía**, la ontología [*del genitivo griego ὄντος: ser y λογία: ciencia, estudio, teoría*] es el **estudio de lo que es en tanto que es y existe**, también llamada la teoría del ser (el estudio de todo lo que es: qué es, cómo es y cómo es posible)
- ▶ En **informática**, una ontología es una estructura completa de datos que contiene la **definición de todas las entidades relevantes y sus relaciones dentro del dominio**. Es el vocabulario con cuyos términos debe ser descrito todo lo demás dentro de un dominio dado, para facilitar la comunicación/compartición de la información entre diferentes sistemas
- ▶ La base de conocimientos (representación estática) se escribe de acuerdo a la ontología (abstracción) adoptada en el dominio de la aplicación

Definición de ontología.



Diferencias entre sistemas basados en conocimiento y sistemas expertos, ya estudiado.

Dendral (E. Feigenbaum, Univ. Stanford, 1965-)

- ▶ Primer sistema “inteligente”, utilizado durante más de 10 años por químicos y biólogos para propósitos reales
- ▶ Inferencia de estructuras moleculares a partir de las medidas del espectrómetro de masas
- ▶ Inicialmente escrito en Lisp
- ▶ No se considera técnicamente el primer SE al no separar de forma explícita el conocimiento del motor de inferencia

Dendral: primer sistema inteligente de la historia, sobre inferencia de estructuras moleculares a partir de medidas del espectrómetro de masas.

MYCIN (E. Shortliffe, Univ. de Standford, 1972-)

- ▶ Considerado el **primer SE de la historia**
- ▶ Sistema para ayudar a los médicos en la investigación y diagnóstico de **enfermedades infecciosas** en la sangre
- ▶ Al ser consultado, intenta establecer una **hipótesis de diagnóstico** para un determinado paciente, a partir de la información que va solicitando al médico según la necesita
- ▶ Incorpora un generador de **explicaciones**

MYCIN, un sistema de ayuda al diagnóstico de enfermedades infecciosas en la sangre, está considerado el primer sistema experto de la historia, puesto que tenía una separación expresa entre el conocimiento y el motor de inferencias.

Reglas

▶ **Aproximadamente 500 reglas como:**

si

la infección es bacteriemia primaria

y la toma del material a cultivar es una toma
estéril

y se cree que la puerta de entrada del organismo es
el tracto gastrointestinal,

entonces

hay bastante evidencia (0.7) de que la identidad del
organismo sea bacteroides

▶ **Codificada en triplas OAV (infección, tipo, bacteriemia)**

MYCIN estaba codificado en reglas como la mostrada, codificadas en triplas (infección, tipo, quién la produce).

Ejemplo de diálogo (1)

M - ¿Nombre del paciente?

U - Juan Gómez

M - ¿Edad?

...

M - ¿Qué organismo es el causante de la infección?

U - No sé

M - ¿Crece el organismo en medio aerobio?

U - Sí (0.8)

M - ¿Crece en medio anaerobio?

U - Sí

M - ¿Cuál supone que ha sido la puerta de entrada?

U - El tracto gastrointestinal (0.6).

...

M - Mis recomendaciones terapéuticas van a basarse en las siguientes infecciones y organismos probables:

Infección-1: Bacteriemia primaria (PSEUDOMAS AERUGINOSA)

...



Universidad
Carlos III de Madrid

IRC 2011/2012 - 44

Ejemplo de un posible diálogo de MYCIN con un médico. Captura de datos y recomendación diagnóstica.

Ejemplo de diálogo (2)

35)M - ¿La meningitis se ha contraído en un hospital?

U - ¿Por qué?

M - Ello me ayudaría a determinar los organismos. Ya he concluido que:

1. La infección es meningitis
2. No se han visto organismos en el teñido del cultivo
3. El tipo de infección es bacteriana
4. No figura neurocirugía en el historial médico y
5. La edad del paciente es superior a 10 días

Por tanto, si

6. El paciente contrajo la infección en un hospital entonces (regla 545) los organismos causantes podrían ser E. Coli (0.75), estafilococo coagulante positivo (0.3), pseudomonas aeruginosa (0.3) o klebsiella pneumoniae (0.5).

Volvamos a la pregunta 35.

U - ¿Cómo 3?

M - Por las siguientes reglas:

Regla 500 (0.51)

Regla 526 (0.98)

...



Universidad
Carlos III de Madrid

IRC 2011/2012 - 45

Continuación del ejemplo de diálogo. Petición de explicaciones por parte del médico:
“¿por qué me preguntas eso?”, “cómo has llegado a esa conclusión?”.

Aplicaciones de los SBC

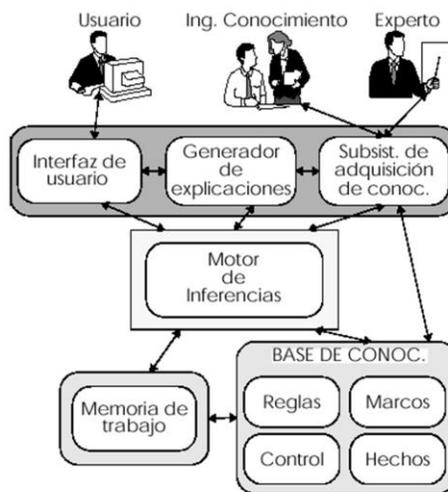
- ▶ **Diagnóstico y reparación de sistemas y/o dispositivos de todo tipo**
 - ▶ Sistemas que diagnostican fallos y sugieren medidas correctoras
- ▶ **Monitorización y control de procesos**
 - ▶ Análisis en tiempo real de dispositivos o procesos físicos, para la detección temprana de anomalías, desviaciones o tendencias, o su gestión óptima
- ▶ **Planificación**
 - ▶ Analizan una serie de objetivos potencialmente complejos e interrelacionados, para determinar el conjunto óptimo de acciones a realizar para conseguir esos objetivos y establecer su planificación en el tiempo, teniendo en cuenta las restricciones existentes
- ▶ **Configuración (fabricación y diseño)**
 - ▶ Selección y disposición de un conjunto de elementos relacionados por una serie de restricciones
- ▶ **Gestión del conocimiento**
 - ▶ Distribución de conocimiento a los usuarios de la manera óptima para satisfacer sus intereses (sistemas de recomendación y de ayuda)

Aplicaciones de los sistemas basados en conocimiento.

Arquitectura (general) de los SBC

- ▶ base de conocimiento
 - ▶ hechos y reglas
 - ▶ independiente del dominio
 - ▶ dependiente del dominio
 - ▶ conocimiento de control
- ▶ motor de inferencias
 - ▶ encadenamiento hacia delante y hacia atrás
- ▶ memoria de trabajo
- ▶ interfaz
- ▶ generador de explicaciones

[También existen otros sistemas basados en redes bayesianas]



Arquitectura general de los sistemas basados en conocimiento: el motor de inferencias, la base de conocimiento (conocimiento estático sobre el dominio, a largo plazo), la memoria de trabajo (conocimiento dinámico sobre el razonamiento actual, a corto plazo), la interfaz de usuario con un sistema de entrada/salida, un editor del conocimiento, y el generador de explicaciones.

Base de conocimiento

▶ Conocimiento

▶ Reglas:

si <condición> entonces <acción>

▶ Hechos:

- ▶ definiciones, tipos, restricciones, valores por defecto...

[CBR: conocimiento de casos anteriores]

▶ Control

- ▶ Estrategia de alto nivel de la aplicación: fases de la aplicación, bases de conocimiento que se van a utilizar

Características de la base de conocimiento, que incorpora reglas y hechos y una estrategia de control [ver Tema 2 de Resolución de Problemas].

El paradigma es el de razonamiento basado en casos (“case-based reasoning”, CBR), es decir, razonar utilizando conocimiento de casos anteriores.

Tipos de conocimiento

- ▶ **Normativo**: general sobre el dominio
- ▶ **Factual**: hechos concretos
(*siempre declarativo*)

- ▶ **Táctico**: reglas de inferencia, metarreglas
- ▶ **Estratégico**: de control y búsqueda
(*en principio procedimental*)



Diferentes clasificaciones del tipo de conocimiento.

Motor de inferencias

- ▶ Colección integrada de algoritmos de resolución de problemas
- ▶ Codificado y probado
- ▶ Funcionamiento del proceso de inferencia:
 - ▶ detección de las reglas aplicables
 - ▶ elección de reglas (resolución de conflictos)
 - ▶ aplicación de la regla
 - ▶ actualización de la base de conocimientos
 - ▶ (repetir el ciclo hasta que no haya reglas aplicables)

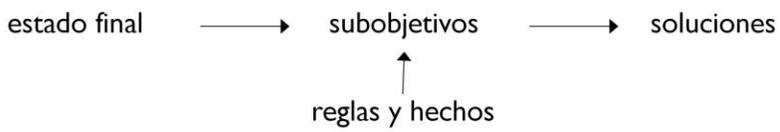
Características del motor de inferencias.

Reglas de inferencia

Encadenamiento hacia delante (*forward chaining*)



Encadenamiento hacia atrás (*backward chaining*)



Reglas de inferencia con encadenamiento hacia adelante (del estado inicial, aplicando reglas, se obtienen conclusiones) o encadenamiento hacia atrás (del estado final, se buscan los hechos y reglas para explicar el punto de partido).

Resolución de conflictos

Cuando hay más de una regla aplicable, hay que elegir

- ▶ **Preferencia basada en reglas**
 - ▶ asignación de prioridades explícita, orden de introducción (Prolog), historia de la regla (*more/less-recently used*)...
- ▶ **Preferencia basada en objetos**
 - ▶ reglas que se aplican a los objetos más recientes, asignación de prioridades a objetos (patrones más comunes)
- ▶ **Preferencia basada en estados (heurísticos)**
 - ▶ banco de pruebas, analizar los resultados, seleccionar la mejor

Al igual que en resolución de problemas, cuando existe más de una regla aplicable, hay que elegir, aplicando una estrategia de resolución de conflictos.