

1. La tarea de diagnóstico. Definición, entrada y salida, método de resolución, diagrama inferencial, algoritmo de resolución.

Definición

El objetivo de la tarea de diagnóstico es encontrar un defecto o avería que provoca una disfunción en un sistema. Por ejemplo, en el dominio técnico, una tarea usual es el diagnóstico de un dispositivo estropeado tal como un coche o un electrodoméstico.

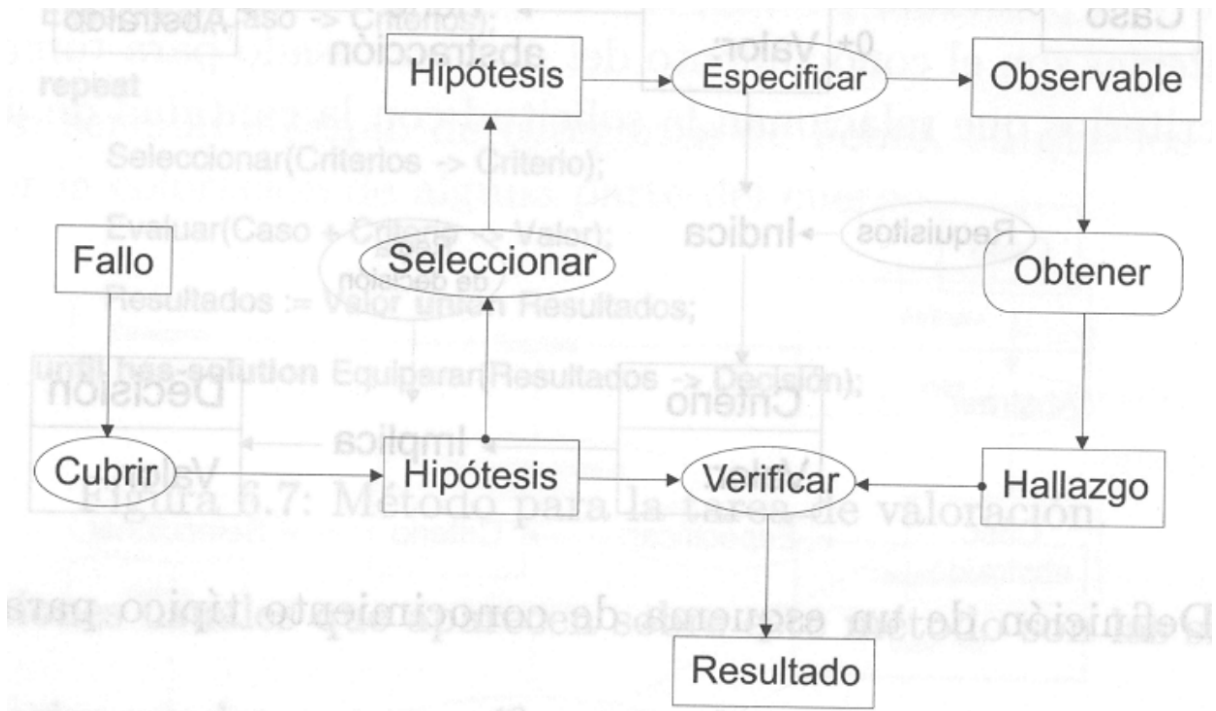
Entrada y salida

Como datos iniciales tenemos las quejas o fallos y como datos de salida las posibles soluciones (hipótesis).

Método de resolución

El método por defecto es el método de cobertura causal, que asume que existe algún modelo causal simple del comportamiento del sistema. A veces, se sustituye por el de asociaciones directas entre fallos y defectos, quedando la tarea de diagnóstico en una tarea de clasificación.

Diagrama inferencias



Algoritmo de resolución

```
while new-solution Cubrir(Fallo -> Hipótesis) do  
  Diferencial := Hipótesis add Diferencial;  
end while  
repeat  
  Seleccionar(Diferencial -> Hipótesis);  
  Especificar(Hipótesis -> Observable);  
  Obtener(Observable -> Hallazgo);  
  Evidencia := Hallazgo add Evidencia;  
  foreach Hipótesis in Diferencial do  
    Verificar(Hipótesis + Evidencia -> Resultado);  
    if Resultado = falso  
      then Diferencial := Diferencial subtract Hipótesis  
  until length Diferencial =< 1 or “no existen más observables disponibles”  
  Defectos := Hipótesis;
```

2. La tarea de valoración. Definición, entrada y salida, método de resolución, diagrama inferencial, algoritmo de resolución.

Definición

Su objetivo es identificar un caso como miembro de una categoría de decisión, teniendo en cuenta criterios específicos del dominio. Por ejemplo, en el dominio financiero, una tarea usual es la decisión de prestar o no un crédito solicitado.

Entrada y salida

Como entrada se recibe el caso a valorar (información relativa a este) y como salida se obtiene la decisión tomada.

Método de resolución

El método de resolución se divide en dos pasos:

1. Abstracter los datos del caso de forma repetitiva hasta que se abstraigan todos
2. Especificar los criterios, seleccionar un único criterio, valorar dicho criterio y ver si con él se llega a alguna decisión. Si no se llega a ninguna decisión, este proceso se repite con todos los criterios hasta que se llegue a la decisión.

Diagrama inferencial

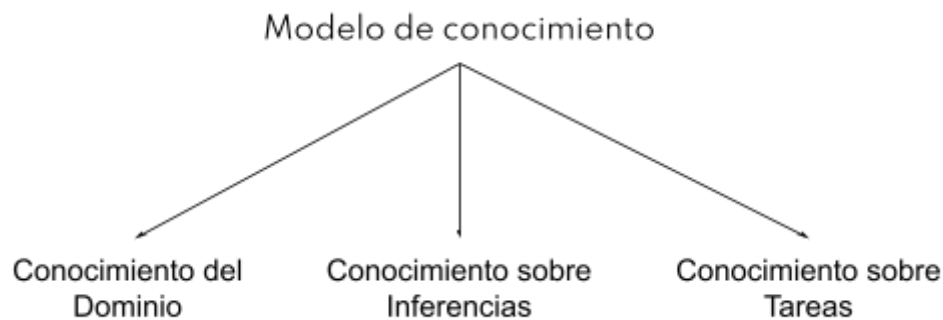


Algoritmo de resolución

```

while new-solution Abstraer(Caso -> Característica-abstraída) do
    Caso := Característica-Abstraída union Caso;
end while
Especificar(Caso -> Criterios);
repeat
    Seleccionar(Criterios -> Criterio);
    Evaluar(Caso + Criterio -> Valor);
    Resultados := Valor union Resultados;
until has-solution Equiparar(Resultados -> Decisión);
  
```

3. Realice un esquema de las categorías del conocimiento y describa la estructura interna de cada una de ellas.



- Conocimiento del dominio:
 - **Esquema del dominio:** Incluye la terminología general utilizada para describir el conocimiento del dominio particular.
 - **Bases de conocimiento:** Amplía el esquema del dominio con hechos específicos de dicho dominio.
- Conocimiento de las inferencias:
 - **Inferencias y roles de conocimiento:** Las inferencias son las unidades básicas de procesamiento de información. Por otro lado, los roles de conocimiento son etiquetas abstractas que indican el papel que juega el conocimiento del dominio en el proceso de razonamiento.
 - **Funciones de transferencia:** nos permite transmitir un elemento de información/conocimiento entre el agente de razonamiento y el mundo exterior.
 - **Diagramas de inferencia:** Especifican las capacidades de razonamiento del sistema que se está desarrollando. Las inferencias se relacionan entre sí mediante sus roles de entrada y salida.
- Conocimiento de las tareas:
 - **Tareas:** Las tareas definen un proceso de razonamiento complejo.
 - **Métodos de las tareas:** El método de una tarea nos indicará cómo se puede alcanzar el objetivo indicado en la tarea, mediante su descomposición en subtareas, inferencias y funciones de transferencia.

4. Enumere y valore las ventajas e inconvenientes del uso de la metodología CommonKads para el desarrollo de un sistema basado en conocimiento.

Ventajas:

- Separación del análisis y la implementación.
- Garantiza la calidad: Al proporcionar la documentación necesaria para su mantenimiento, permitir estudiar la viabilidad y costes del sistema y permitir organizar la programación del sistema.
- La especificación del diseño es independiente a la plataforma donde se va a implementar.
- Facilita la descomposición de tareas.
- Facilita la reutilización.

Inconvenientes:

- Es una metodología muy compleja y amplia
- Mucha información con difícil acceso, comprensión y aprendizaje
- No hay una fuente de información que contenga todo lo necesario para su aplicación
- Sigue un proceso de desarrollo tradicional, como la metodología en cascada, que puede no ser apto para todos los casos.