

# Tema 6 - Verificación y Validación de SBCs

Profesor Juan Luis Castro Peña, Curso 20-21

## Principales errores en el desarrollo de un Sistema Basado en el Conocimiento:

- **Conocimiento/Experto:** Errores de conocimiento del Experto, tales como conocimiento incorrecto o incompleto.
- **Ingeniero del Conocimiento:** Errores semánticos de significados entre el ingeniero de conocimiento y el Experto. Obtención incompleta del conocimiento proveniente del Experto.
- **Base de Conocimiento:** Errores de sintaxis, errores de contenido debido a un conocimiento incorrecto e incompleto y a la incertidumbre en reglas y hechos.
- **Motor de Inferencia:** Errores de la programación y errores lógicos.

### Calidad de un Sistema Experto

- Un SE de calidad cuenta con...
  - A alto nivel
    - Conclusiones correctas
    - Un desempeño adecuado.
    - Código comprensible y comentado.
  - Para llegar a estas condiciones de alto nivel, se deben cumplir estas condiciones de bajo nivel:
    - Conclusiones completas y congruentes: Deduce todo lo que sea necesario y que tiene sentido
    - Que sean confiables respecto a la conclusión: Con datos de entrada parecido, las conclusiones deben de ser parecidas también.
    - Disponibilidad.
    - Una base de conocimiento verificada.

### Funcionalidad de un SBC

- La funcionalidad del SBC es la capacidad del sistema para hacer el trabajo para el que fue destinado.
  - Debe cubrir las expectativas para lo que fue construido.
  - Debe ser confiable respecto a su funcionamiento, da razonamientos lógicos.
  - Debe explicar sus respuestas.
  - Debe permitir que se añada o modifique su conocimiento fácilmente.

### Eficiencia y Errores de un SBC

- Un SBC se considera correctamente validado cuando cubre los requisitos de **calidad** y **funcionalidad**, con lo cual se asegura la aceptabilidad del SBC.
- Un SBC aceptable es un sistema "completo" y eficiente, para todos los casos que se ha diseñado sea capaz de responderlo.
- Si en el proceso de verificación y validación se comenten errores, esto nos trae como consecuencia un SBC mal estructurado y por lo tanto, un sistema con errores e incompleto.
- Para que un SBC o SE sea una herramienta efectiva, los usuarios deben interactuar de una forma fácil, reuniendo dos capacidades para poder cumplirlo.
  - Explicar sus razonamientos y la base de conocimiento
  - Adquisición de nuevos conocimientos o integrador del sistema: Son mecanismos de razonamiento que sirven para modificar conocimientos anteriores.

## Verificación y Validación:

### Diferencias con Ingeniería del Software

- En Ingeniería del Software, se realizan casos de prueba para verificar un programa.
- En Ingeniería del Conocimiento:

- Los criterios para medir el éxito no son objetivos, es cuestión de percepción.
- Se tolera la incertidumbre y la subjetividad.
- No se pueden probar fácilmente: Los espacios de búsqueda son muy grandes por ser problema de IA.
- En muchos casos no existen respuestas "absolutamente correctas" para evaluar el sistema.

### Verificación

- Consiste en construir el sistema correctamente
- Descubrir y corregir los errores en el SBC de naturaleza técnica (El sistema y/o conocimiento no resulta coherente, no se analiza si es válido, solo si es coherente.)
- La realiza el **Ingeniero de Conocimiento**, el Experto participará en la fase de validación donde se analiza si el conocimiento es válido y suficiente para el problema abordado.
- Criterios para verificar un SBC:
  - **Consistencia:** No se llegan a conclusiones incoherentes.
  - **Corrección:** Hay corrección en la sintaxis, no hay errores morfológicos.
  - **Compleitud:** No hay lagunas en la capacidad deductiva, estos son casos donde el sistema no daría respuesta.

### Tipos de Inconsistencias

- **Estructural:**
  - Si posee reglas inútiles, ya sean inalcanzables, que llevan a cajellones sin salida, no sean ejecutables o redundantes, no disparables.
  - Que se produzcan ciclos de reglas, típico en reglas que modifican su antecedente.
- **Lógica:**
  - Reglas con conclusiones o antecedentes redundantes.
  - Subsunción de reglas, es decir, reglas que ya están incluidas en otras y si se quitan el sistema funciona igual.
  - Reglas con conclusiones que producen contradicciones lógicas, y que se podrían ejecutar en una misma situación. Puede existir inconsistencia teórica pero que no se dé en la práctica.
- **Semántica:**
  - Valores ilegales en las variables, se necesita un modelo de coherencia para definir conflictos semánticos porque dependen del contexto.
    - Coherencia de un patrón o conjunto de patrones.
    - Coherencia de una regla o conjunto de reglas.

### Validación

- Significa construir un sistema "correcto", determinar que un sistema satisface las necesidades del usuario y el Experto, por lo que la validación se realiza junto a ellos.
- Se realiza la validación en cada parte del sistema en que se puede aplicar cuando se termine, y al final con todo el sistema en general.
- **Tipos:**
  - **Objetiva:** Basada en especificaciones formales.
  - **Interpretativa:** Actividades encaminadas a eliminar los errores de tipo conceptual y de contexto, a veces denominada evaluación.

### Cumplir con las especificaciones del modelo de diseño

- La representación elegida y la técnica de razonamiento asociada a esa son adecuadas.
- Se refleja el modelo conceptual en la implementación.
- El diseño y la implementación se ha pensado en la modularidad.
- La comunicación entre los subsistemas es adecuada y el sistema en si es fácil de mantener y comprender.

### Aspectos de la Validación

- **¿Qué se está validando?**
  - La comunicación del sistema con otros sistemas es adecuada.
  - La interfaz es comprensible para el usuario.
  - La explicación del razonamiento del sistema es suficiente.

- Cumple los requisitos de ejecución en tiempo real pedidos.
- El sistema cumple con las especificaciones de seguridad.
- Satisfacción y utilidad de los resultados finales e intermedios comparándolos con resultados conocidos, lo que sabe el Experto o un modelo algorítmico.
- **Metodología de la Validación**
  - **Informal:** Reuniones con usuarios y expertos
  - **Mediante casos de prueba:** Tener un conjunto de casos a probar, aquellos que serían los usuales.
  - **Pruebas de Campo:** Actuación en paralelo con el Experto.
  - **Validación de Subsistemas:** Muy importante, imprescindible en sistemas mínimamente complejos.
  - **Análisis de Sensibilidad para Sistemas con Incertidumbre:** Los cambios pequeños provocados por los posibles casos que se puedan dar en la incertidumbre no deberían dar resultados muy dispares.
- **Criterios de Validación**
  - Cuántos casos de prueba.
  - Cómo se generan los casos de prueba.
  - Establecer una proporción entre casos de prueba fáciles, medios y difíciles.
  - Como comparar los resultados con el de un Experto.
  - Cómo se mide el desempeño de un Experto en ese campo.
  - Cómo evaluar el sistema cuando distintos Expertos opinan diferentes cosas.
- **Resultados del Proceso de Validación**
  - Exactitud y Aceptabilidad de las Soluciones
    - ¿Cuántas veces acierta? De acuerdo al criterio prefijado para ello.
  - Adecuación al problema
    - ¿Cubre el dominio? ¿Responde ante todos los casos reales que se puedan dar?
  - Errores:
    - Por comisión: Se responde incorrectamente.
    - Por omisión: No se da respuesta.

#### **Pasos de Verificación y Validación:**

- **Verificar** si el sistema es completo, correcto y consistente.
- **Evaluar** si el sistema cumple especificaciones del modelo de diseño.
- **Diseñar** un plan de validación aplicando metodologías apropiadas.
- **Valorar** en función de criterios de validación entre los otros requisitos funcionales definidos en la fase de identificación del problema.