

Modelos de Inteligencia Artificial

*Conforme a contenidos del «Curso de Especialización
en Inteligencia Artificial y Big Data»*



**Modelos de
Inteligencia_Artificial**

Universidad de Castilla-La Mancha

Escuela Superior de Informática
Ciudad Real

6

Capítulo

Sistemas Expertos (I): Conceptos y metodología

José Angel Olivas Varela

Para aprovechar datos de forma inteligente, casi siempre conviene, si es posible, contar con un experto o varios para que ayuden a “interpretar” la semántica, limitaciones, calidad, objetivos del análisis... de los datos a analizar. Esta interacción con los expertos en los diferentes temas está profusamente estudiada y descrita en la disciplina que suele denominarse “Ingeniería del Conocimiento”, que en general se encarga del desarrollo de los Sistemas Basados en Conocimiento (SBS) y en particular de los llamados “Sistemas Expertos”. Por lo general, el desarrollo de un SBC suele ser, en parte, similar al de otros programas convencionales. Por ello, se podría decir que hay una parte donde surgen problemas y dificultades relacionadas con el contexto de los SBC y otra parte en la que las dificultades son comunes al desarrollo convencional de software. No obstante, se va a hacer hincapié en el ámbito que nos ocupa, y así tratar de describir las sucesivas fases que integran el proceso de desarrollo de un SBC, fases que, al estar tan interrelacionadas entre sí, no siempre es posible establecer una clara separación entre las mismas. No existe una metodología única de desarrollo para los distintos tipos de Sistemas Basados en Conocimiento. Sin embargo, puede resultar interesante la elaboración de un esquema genérico aproximado, cuya estructura y orden podría variar dependiendo fundamentalmente del objetivo del programa a diseñar.

En el desarrollo de un SBC siempre se tiene en cuenta la interacción entre los Ingenieros de Conocimiento y los expertos humanos en el dominio. Ahora bien, la actuación de ambos no debería ignorar las aspiraciones y necesidades de una tercera entidad en juego que no es otra que los destinatarios o usuarios finales del programa. El papel protagonista del desarrollo suele corresponder a los dos primeros, pero el éxito o fracaso del sistema dependerá en gran medida de las aportaciones y colaboraciones que se hayan dado entre las tres entidades involucradas.

6.1. Inteligencia Artificial e Ingeniería del Conocimiento.

La Ingeniería del Conocimiento es una disciplina que forma parte de la **Inteligencia Artificial** centrada en el análisis y propuesta de métodos para:

- **adquirir** el conocimiento humano
- **representar** el conocimiento adquirido
- **guardar** ese conocimiento
- usar el conocimiento en un proceso de **razonamiento**

Todo ello con vistas a emular determinadas capacidades inteligentes propias del ser humano.

Sistemas Expertos

Un **Sistema Experto** es un programa de ordenador que trata de emular el comportamiento de una persona experta en un dominio de conocimiento específico ante un problema que se plantee en dicho dominio y cómo llega a su solución.

Ejemplo: Un médico realizando su trabajo.

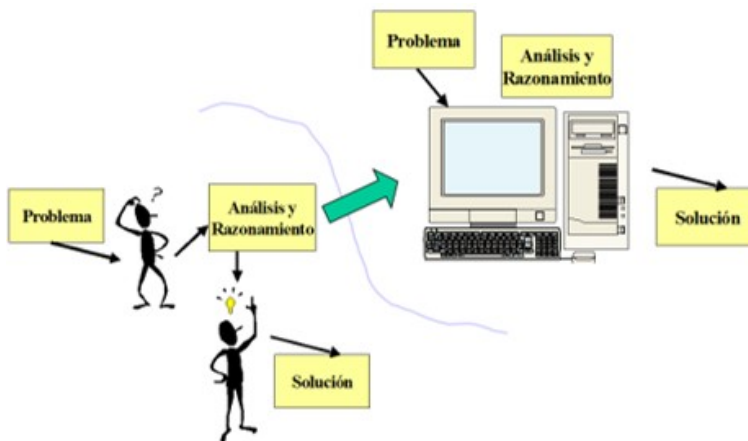


Figura 6.1: Símil de la actuación de un experto y un sistema experto.

La **Ingeniería del Conocimiento** se ocupa entre otras cosas del proceso de especificación, análisis y desarrollo de un sistema experto.

La Idea origen se puede encontrar en el nacimiento de las primeras máquinas capaces de procesar números y símbolos. (Babbage 1842, Turing 1936). Las principales contribuciones al desarrollo de la idea han sido:

6.1. Inteligencia Artificial e Ingeniería del Conocimiento.

- **Teóricas:** El desarrollo de los fundamentos de la lógica proposicional (Boole, 1854) y de predicados de primer orden (Frege, 1879) y las ideas de Alan M. Turing (1950) sobre posibilidad de mecanizar la inteligencia humana.
- **Tecnológicas:** Primeros ordenadores en los 40.

Es frecuente señalar que la IA nació en el verano de 1956 en un workshop en el Dartmouth College en New Hampshire. Esta conferencia fue organizada por John McCarthy, Marvin Minsky, Nathaniel Rochester y Claude Shannon. John McCarthy, el creador del lenguaje LISP, propuso el término **Inteligencia Artificial** que se ha venido utilizando hasta hoy.

Entusiasmo inicial (1956-1969). En la década de los 60, Herbert Simon, Allen Newell y Cliff Shaw hicieron experimentos escribiendo programas para imitar el proceso de razonamiento humano. Construyeron un programa llamado “General Problem Solver (GPS)” que contenía reglas de transformación de expresiones lógicas. Cuando se le introducía una expresión lógica, buscaba a través de todas las posibles reglas, la demostración de la verdad o no de la expresión introducida. En IBM, Rochester y sus colegas desarrollaron algunos de los primeros programas de IA. Gelernter construyó un demostrador de teoremas de geometría. Samuel escribió una serie de programas de juegos de damas (nivel amateur) y Shannon escribió un artículo sobre la posibilidad de que los ordenadores jugaran al ajedrez.

Sistemas basados en conocimiento (1969-1979). Tras los primeros desarrollos basados en procedimientos de búsqueda general para resolver problemas, llegó el obstáculo cuando había que enfrentarse a casos concretos de mediana complejidad. Aparece la necesidad de usar conocimientos específicos del dominio tratado y la necesidad de crear estrategias de razonamiento en varias etapas. Surgen los Sistemas Expertos.

La IA pasa a la industria (1980-hasta hoy). El primer sistema experto comercial fue R1 (1982), construido por el fabricante DIGITAL. Permitía la elaboración de pedidos de nuevos sistemas informáticos. En 1986 ahorra a la compañía 40 millones de dólares al año. Hubo un crecimiento espectacular y a veces se exageraron las expectativas. Hoy día múltiples herramientas de IA están en la industria.

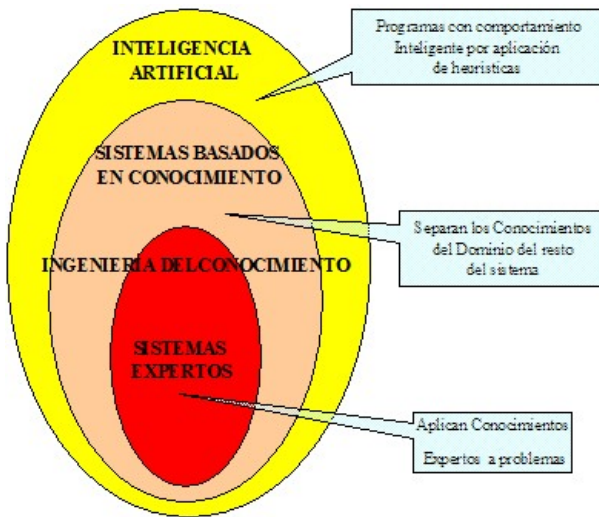


Figura 6.2: IA, IC y Sistemas Expertos.

Las principales ventajas del uso de sistemas expertos son:

- Ayuda a personas no expertas a resolver problemas en un dominio de conocimiento.
- Ayuda a las personas expertas mediante contraste de sus diagnósticos.
- Entrenamiento y aprendizaje de no expertos.
- Documentación, acumulación y distribución del conocimiento más eficiente y amplia capacidad de la base de conocimientos.
- Posible revisión y mejora del conocimiento.
- Razonamiento sobre todo el conocimiento.
- Fácil seguimiento de los pasos dados en la resolución de problemas. Fácil seguimiento del razonamiento.
- Programas de utilización sencilla, flexible y de fácil mantenimiento y actualización.

Se puede detectar la necesidad de disponer de un sistema experto cuando se da alguna de las siguientes situaciones en una empresa u organización:

- Cuando un reducido número de personas clave en una organización dedica una parte importante de su tiempo ayudando a otros.
- Cuando la realización de tareas aparentemente sencillas requiere un equipo humano numeroso porque ninguno de los componentes tiene el conocimiento suficiente completo.

6.1. Inteligencia Artificial e Ingeniería del Conocimiento.

- El rendimiento en la realización de una tarea se degrada porque requiere un análisis pormenorizado de un juego complejo de condiciones y el ejecutor típico nunca parece recordarlo todo.
- Cuando se presenta una amplia diferencia entre el mejor y el peor de los realizadores de una tarea.

Las principales razones que puede haber para desarrollar un sistema experto en una organización o empres pueden ser:

- Preservar el conocimiento que puede perderse por baja de un experto de reconocido prestigio en la organización en que trabaja.
- Replicar las habilidades de un experto en un programa para que su conocimiento pueda ser diseminado o utilizado de forma distribuida.
- Guardar el conocimiento en una forma activa (Base de Conocimientos) mejor que en una pasiva (libro, manual).
- Proporcionar a los menos expertos una herramienta que les ayude a pensar de un modo semejante a como lo haría un experto.
- Disponibilidad permanente del conocimiento sin estar sujeto a fallos humanos (fatiga, estado emocional...).

Los principales requisitos deseables en un sistema experto son:

- Debe resolver un problema que de otro modo hubiera debido ser resuelto por un experto humano.
- Debe ser flexible en integrar nuevo conocimiento incrementando el existente en su base de conocimientos.
- Debe asistirnos en las tareas de obtener, estructurar y transferir conocimiento.
- Debe ser capaz de presentar su conocimiento de forma que sea fácil su lectura.
- Debe proporcionar explicaciones sobre sus conclusiones.
- Debe ser capaz de razonar con conocimiento impreciso e inexacto.
- Debe ser capaz de trabajar con frases sencillas en lenguaje seminatural.

Las principales limitaciones en la construcción de sistemas expertos vienen dadas porque el conocimiento experto humano es experiencia compilada, es Heurístico, esto es, basado en experiencia, reglas prácticas. Es extenso y también redundante e incluye heurística acerca de la heurística y del razonamiento. Es incompleto, impreciso e incierto, (a veces inconsistente y con errores o imprecisiones). Es por ello que las limitaciones de todo SE pueden ser que no conoce lo que conoce ni por qué, carece de imaginación, emociones, inteligencia innata, sentido común, etc. Tiene poco conocimiento de sí mismo, del usuario y del contexto de cada interacción y capacidad de razonamiento limitada por su estrategia de construcción.



Figura 6.3: Primeros sistemas expertos muy conocidos y exitosos.

Hoy en día, los sistemas expertos suelen estar incorporados en el entorno de aplicaciones de las empresas y entidades, pero las principales herramientas de desarrollo que se suelen usar son:

- **CLIPS** (C Language Integrated Production System). Entorno basado en reglas de razonamiento hacia delante escrito en C por NASA.
- **FuzzyCLIPS** CLIPS para implementar Lógica Borrosa o Difusa (Fuzzy Logic, Razonamiento aproximado).
- **Lenguajes de programación** de propósito general (con extensiones como CLIPSPY...).

6.2. Metodología y ciclo de vida en el desarrollo de un sistema experto.

Un desarrollo metodológico en ingeniería exige la definición y estandarización de un ciclo de vida desde la especificación de requisitos hasta el mantenimiento del producto final. Esta exigencia está perfectamente definida en la Ingeniería del Software (IS) pero tiene lagunas en la Ingeniería del Conocimiento, debido fundamentalmente a la ausencia de requisitos, cosa fundamental en la IS. La ausencia de requisitos en la Ingeniería del Conocimiento se manifiesta de 3 maneras diferentes:

- Se tratan problemas mal estructurados y mal definidos, lo que dificulta un análisis de requisitos y su especificación.
- La elaboración de SE es un proceso de mejora gradual y continuo que impide dejar cerrados ciertos aspectos de la aplicación.
- Dado que se quiere modelar el comportamiento de los expertos y, éstos por regla general no tienen claro los requisitos de su quehacer, difícilmente se puede realizar una especificación de requisitos a priori.

6.2. Metodología y ciclo de vida en el desarrollo de un sistema experto.

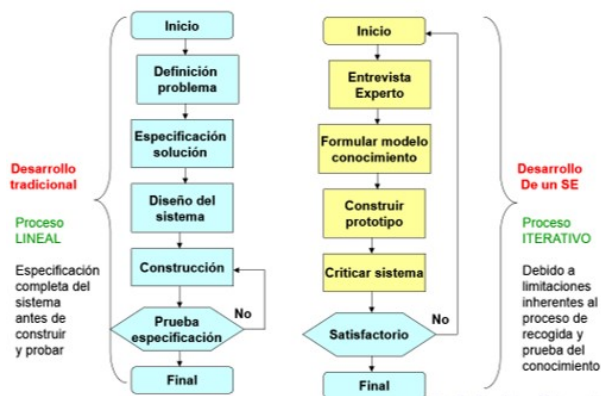


Figura 6.4: Desarrollo tradicional vs. desarrollo de un sistema experto.

La manera de conciliar esa discrepancia entre IS e Ingeniería del Conocimiento es mediante el empleo de una Metodología que proponga un cierto conjunto de pautas, suposiciones o principios a seguir en el desarrollo de un SE. Esta metodología debe proporcionar claramente un conjunto de criterios para identificar, ordenar, evaluar y usar métodos, técnicas y herramientas adecuados para el desarrollo de un SE. Por otro lado, una metodología describe cómo se organiza un proyecto, el orden en que se ejecutan las tareas y los interfaces entre ellas. El desarrollo de la metodología da lugar al modelo de Ciclo de Vida que indica las actividades que se han de realizar en cada fase del mismo. Las principales ventajas de dividir el proyecto en fases son que se puede evaluar el progreso del proyecto por los resultados obtenidos al final de cada fase y que facilita la estimación de medios y tiempos necesarios en cada fase. Se pueden establecer tres generaciones diferentes de metodologías de desarrollo de SE:

- Ciclo de vida del tipo 'Codifique y arregle'. Es la forma más primitiva en donde lo importante es implantar los conocimientos con un método de razonamiento y ver si funciona.
- Ciclo de vida en 'Cascada'. Aquí ya hay una retroalimentación entre las distintas fases y sus etapas.
- Ciclo de vida en 'Espiral'. Se componen de varios subciclos de etapas similares, pero en cada subciclo se van añadiendo requisitos y perfecciones sobre lo que hubo en el anterior.

La diferencia entre el método en cascada y en espiral radica en que en el primer caso se ha de desarrollar cada fase del proyecto de manera completa y si se detecta algún error se rehace la fase. En el caso del ciclo de vida en espiral, cada fase se da por acabada parcialmente en un subciclo pero puede mejorarse en el siguiente añadiendo o perfeccionando otros aspectos sin que se vea comprometido el desarrollo hasta ahora hecho. Es un desarrollo progresivo y paralelo de cada fase del ciclo de vida.

Una metodología de construcción de Sistemas Expertos debe contener de forma genérica las siguientes cuatro etapas: Identificación de la tarea, Desarrollo del/os prototipo/os, Construcción y Ejecución del sistema completo, Actuar para conseguir el mantenimiento del sistema, y Lograr una correcta transferencia de tecnología, Metodología IDEAL propuesta por Alonso y Juristo, 1995:

1. **Identificación de la tarea:** Consiste en identificar el problema y definir los objetivos a conseguir para ver si la tarea es susceptible de ser tratada por la IC. Si lo es:

- Se definen las características del problema.
- Se especifican los requisitos que enmarcan la solución del problema
Esta fase se divide en las siguientes tres etapas:
 - Plan de requisitos y Adquisición del conocimiento inicial.
 - Evaluación y selección de la tarea. Viabilidad.
 - Definición de las características de la tarea y análisis de riesgos.

2. **Desarrollo del/os prototipo/os:**

- Diseño general del sistema prototipo a partir de resultados de etapas anteriores.
- Adquisición más profunda del conocimiento y conceptualización del mismo.
- Formalización del conocimiento: Selección de formalismo de representación y diseño detallado de la arquitectura computacional.
- Implementación y selección de Herramientas.
- Validación y evaluación.
- Definición de nuevos requisitos, especificaciones y diseño.

3. **Construcción y ejecución del sistema completo:**

- Expansión de las Bases del Conocimiento.
- Establecimiento de las interfaces con otros sistemas hardware, software y otros, y de la transportabilidad del sistema.
- Adecuación de la interfaz de manejo al usuario final.
- Monitorización de las prestaciones y evaluación del rendimiento del sistema completo.
- Prueba y mejora de las prestaciones del sistema final global.
- Aceptación por el usuario.

4. **Actuar para conseguir el mantenimiento del sistema, y lograr una adecuada transferencia de tecnología:**

6.2. Metodología y ciclo de vida en el desarrollo de un sistema experto.

- Definir el mantenimiento del sistema en general y de las bases del conocimiento en particular.
- Organizar la transferencia tecnológica.
- Completar y finalizar la documentación del Sistema Experto construido.

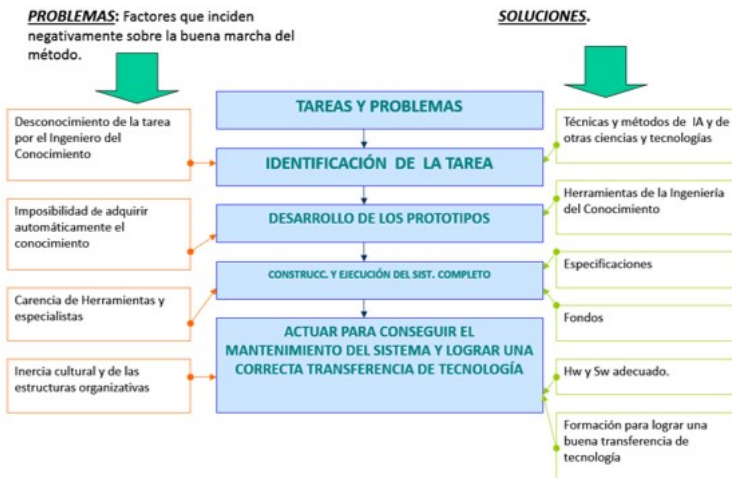


Figura 6.5: Problemas y posibles soluciones en la metodología de desarrollo de un sistema experto.

Desde el punto de vista tecnológico, los SE pueden presentar varias formas de uso:

- **Aislada:** Un SE único se relaciona con el entorno.
- **Integrada:** Varios SE conectados a bases de conocimiento comunes. Tipo Front-End cuando todos los hechos y datos están físicamente en la base común. Tipo Back-End cuando los hechos y datos necesarios que no estén en las bases comunes, pueden obtenerse de otros SE.
- **Embebida:** Un SE está integrado con otros sistemas y no se distingue.

Es frecuente utilizar en el desarrollo técnicas combinadas de Rápido Prototipaje y Desarrollo Incremental. Puede utilizarse la flexibilidad y potencia de CLIPS, LISP, PROLOG o de otras herramientas de desarrollo para crear rápidamente un prototipo funcional del sistema final deseado. Esto permitirá realimentar en forma temprana aspectos de profundidad y tipo de conocimiento, de las necesidades de los usuarios, así como ayudará a verificar la validez de las decisiones tomadas durante la etapa de diseño. Así, si se presenta un dilema de cambio, su impacto será mínimo debido a lo temprano de su ocurrencia.

El prototipo inicial puede ser desechado una vez que se ha obtenido la realimentación esperada o puede ser mejorado en forma incremental para constituirse en un subsistema del SBC final. En la mayoría de los casos se opta por desecharlo debido a las dificultades que existen para modificarlo. En general, es más fácil empezar un nuevo proceso de desarrollo, siguiendo las diferentes etapas del ciclo de vida con mayor detalle. Una vez que el prototipo del sistema ha sido evaluado, se realiza su diseño final y se le somete a un ciclo continuo de modificaciones para que mejore sus características y su calidad. La estrategia que se sigue puede ser definida como un proceso iterativo de extracción, representación y confirmación del conocimiento en un limitado subconjunto del dominio del problema con el objetivo de construir en forma incremental el SBC en segmentos autocontenidos. En consecuencia, el desarrollo incremental involucra varios ciclos de recopilación del conocimiento de los expertos, incorporación del conocimiento en el sistema, revisión de la implementación resultante con los expertos y corrección de los problemas encontrados.

El desarrollo incremental se centra en dos conceptos: Dividir para conquistar, donde un segmento de conocimiento manejable pero completo, es seleccionado y desarrollado y desarrollo iterativo, donde el concepto de dividir para conquistar es aplicado iterativamente sobre los diferentes segmentos que conforman el problema completo

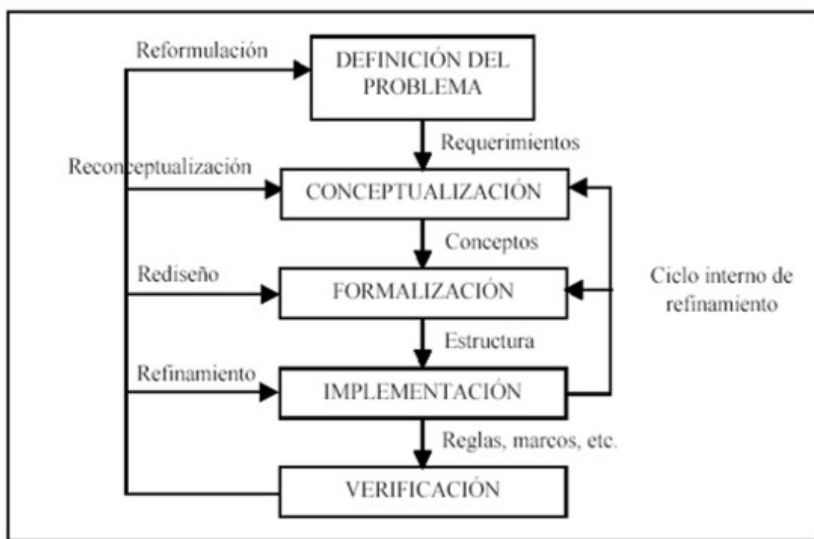


Figura 6.6: Desarrollo iterativo de un sistema experto.

Un **prototipo** es un sistema representativo del problema a resolver. Esta es una etapa clave debido a que todas las decisiones tomadas en el diseño preliminar deben ser confirmadas, rectificadas o desechadas, sobre la base del conocimiento recopilado de los expertos en el proceso hasta completar esta etapa. El prototipo inicial debe verse como el sistema completo, excepto que estará limitado en su cobertura.

6.2. Metodología y ciclo de vida en el desarrollo de un sistema experto.

Debe incluirse una relativamente bien definida interfaz con el usuario y un robusto subconjunto de conocimiento de tal forma que los usuarios puedan juzgar su aceptabilidad. En general se recomienda que el prototipo inicial sea desechado una vez que se haya completado su evaluación.

Las razones para construir un prototipo de Sistema Experto son:

- **DEMOSTRACIÓN:** Sólo se estudia una parte del problema, pero se hace con profundidad, con el objetivo de demostrar su viabilidad.
- **EXPERIMENTACIÓN:** Se estudian diferentes alternativas, sin profundizar demasiado, con el objetivo de investigar distintas alternativas de diseño.
- **EVALUACIÓN.**

Las fases de producción de un sistema completo son las siguientes:

- Implementación del núcleo del sistema completo.
- Expansión de la base de conocimientos anterior.
- Adecuación de la interfaz a las necesidades del usuario.
- Monitorización de prestaciones y evaluación del rendimiento.
- Evaluación del sistema. (Validación y verificación).

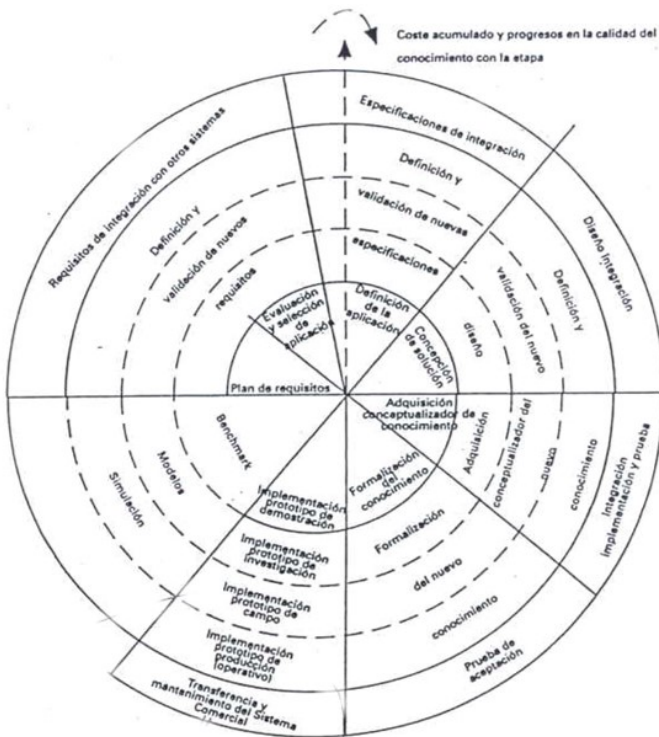


Figura 6.7: Modelo espiral de ciclo de vida de desarrollo de un sistema experto.

En el modelo de ciclo de vida en espiral de tipo cónico, se comienza por el centro de la espiral, se evalúa la viabilidad de la aplicación, se define, adquiere, conceptualiza y formaliza el conocimiento hasta llegar al desarrollo de un primer prototipo de DEMOSTRACIÓN. Después se entra en un ciclo de iteraciones para construir los otros diferentes prototipos (INVESTIGACIÓN y CAMPO). La última vuelta se centra en la integración del último prototipo operativo con los diferentes sistemas de su entorno para conseguir el sistema completo. El desarrollo conlleva el uso de diferentes prototipos, partiendo de la definición inicial del sistema y refinando y añadiendo nuevo conocimiento hasta lograr un SISTEMA completo. Cada uno de estos prototipos genéricos pueden a su vez generar en su evolución varias versiones reales. Los prototipos se desarrollan en este orden:

- **Prototipo de DEMOSTRACIÓN.** Es el primero y más simple. Habitualmente sólo consta de un interfaz y una demo fija. Su utilidad es tanto para hacerse una idea del futuro sistema como para su “promoción” ante posibles interesados (directivos y/o usuarios).
- **Prototipo de INVESTIGACIÓN.** Es el segundo a desarrollar. Habitualmente no se cuida el interfaz (o simplemente no tiene) y se usa como banco de pruebas de algoritmos y procesos. Su utilidad es el diseño y prueba en profundidad de los mecanismos que utilizará el sistema experto.

- **Prototipo de CAMPO.** Es el tercero por desarrollar. Fusión del interfaz del prototipo de demostración y de los mecanismos del de investigación. Es el primer prototipo que suele dar idea real de cómo será el sistema completo. Suele ser usado resolviendo casos en paralelo con los expertos para comprobar las prestaciones y corregir errores.
- **Prototipo de PRODUCCIÓN.** Es el cuarto a desarrollar. Último refinamiento del prototipo de campo. Es prácticamente el sistema completo aunque todavía funcionando de manera aislada del entorno donde será utilizado. Cuando este prototipo se integra con los demás elementos del entorno de utilización se obtiene el SISTEMA COMPLETO.

Como resumen final, como era de esperar, hay una proyección entre ambos conceptos: por Metodología se entiende las tareas “conceptuales” a realizar y por Ciclo de Vida la secuencia temporal real del desarrollo del sistema. En la figura 8 se muestra esta proyección.

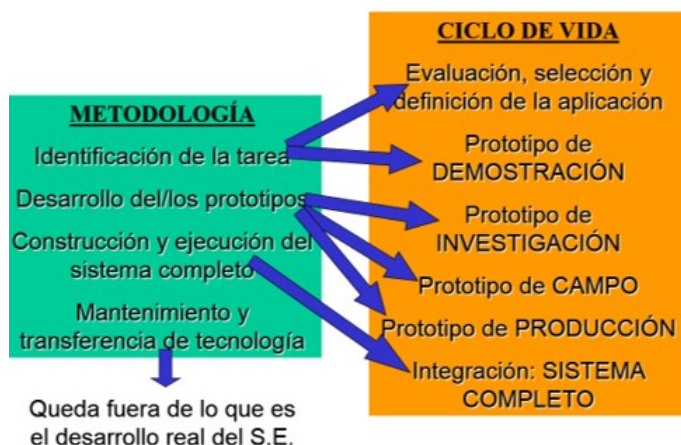


Figura 6.8: Relación entre metodología y ciclo de vida en el desarrollo de un sistema experto.

6.3. Fases de desarrollo de un sistema experto.

A continuación se describen brevemente las fases genéricas para el desarrollo de un SBC:

6.3.1. Etapa 1: Definición del problema. Identificación

En un primer momento, previo al inicio del proceso de desarrollo de la aplicación informática, es sumamente importante realizar una descripción lo más detallada posible de la cuestión que se va a intentar resolver. Es muy conveniente indicar la misión, objetivos, cuáles van a ser las entradas al sistema, y especificar claramente las salidas del mismo. Es decir, se necesita aclarar desde un principio la situación de partida y el nivel de detalle al que se desea llegar.

Si esta fase no queda definida y acotada con suficiente claridad, es posible que al finalizar el desarrollo el usuario del sistema no quede satisfecho porque lo que está obteniendo como resultados, no le sirven para nada. Por tanto, si desde el principio, se tiene una visión muy clara de cuál es el problema y qué resultados se desean obtener, todo será mucho más fácil tanto para el usuario como para el equipo encargado de realizar el SBC.

6.3.2. Etapa 2: Estudio de viabilidad

Lo que se pretende es comprobar si realmente el SBC va a poder contribuir, de una manera eficiente, a la resolución de un problema permitiendo alcanzar todos los objetivos previstos. No siempre la mejor alternativa para resolver convenientemente un problema tiene que ser de forma obligada un SBC o expresándolo de una manera más genérica, las técnicas de Inteligencia Artificial. Son múltiples los métodos que se pueden aplicar con el fin de realizar este estudio, pero quizás uno de los más sencillos y precisos es el método o test de Slagel de 1989.

6.3.3. Etapa 3: Adquisición del conocimiento

“Adquisición de Conocimiento” es la extracción o recogida del conocimiento desde la fuente experta (experto, libros, datos, etc.) por parte del Ingeniero del Conocimiento con objeto de que pueda ser usado por un programa.

Es un proceso muy difícil de intensa interacción que tiene por fin extraer el conocimiento heurístico y experiencia de resolución de problemas del dominio que interesa a partir de las fuentes donde se encuentre.

Aunque no de forma exclusiva, casi siempre que dentro de este ámbito se hace referencia expresa a las posibles fuentes de las que se extraerá la experiencia e información, se suele pensar en un especialista o experto como protagonista fundamental del proceso. De hecho, no parece factible plantear la creación de un programa que acumule experiencia y conocimientos pertenecientes a un campo específico del saber sin tener en cuenta, desde el principio, las fuentes de información provenientes del o los expertos. La elección de un buen experto como fuente inicial de conocimiento es de suma importancia ya que cuanto mayor sea la calidad de su experiencia mejor tenderá a ser la calidad de los resultados del futuro SBC.

No obstante, se puede disponer del mejor experto del mundo en una materia, pero de nada sirve eso si carece de la capacidad y transparencia necesarias para la transmisión de ideas, argumentos, opiniones, creencias, razonamientos, pensamientos, intuiciones, etc. El Ingeniero del Conocimiento tendrá entonces que ingeniárselas para poder sonsacar la información que necesita del experto. Existen varias técnicas para conseguirlo.

La Adquisición de Conocimientos consiste en la recolección de la información necesaria para construir un Sistema Basado en el Conocimiento a partir de cualquier posible fuente. Esta información puede estar constituida por datos, noticias, conocimientos humanos, etc.

La Adquisición de Conocimientos no debe ser considerada como una etapa dentro de una metodología para la construcción de un Sistema Basado en el Conocimiento sino un proceso paralelo a todas las fases de desarrollo de uno de estos sistemas, ya que cada etapa necesita determinada información, lo que provoca que la recolección de ésta no se haga en un único paso aislado sino en cada una de las etapas. El papel de la Adquisición del Conocimiento en las primeras fases (definición del problema, conceptualización, ...) es fundamental, mientras que en las últimas etapas del desarrollo (implementación, evaluación, mantenimiento) la dedicación a la Adquisición de Conocimiento es mucho menor.

Es posible que la tarea más importante para el desarrollo de un Sistema Basado en el Conocimiento sea la Adquisición del Conocimiento. Pero paradójicamente, este es un campo experimental más que una tecnología, y salvo en el caso de la inducción y aprendizaje automático, la Inteligencia Artificial no aporta métodos completos que solucionen o automaticen esta tarea, solo técnicas para abordar problemas parciales. Por consiguiente, la Adquisición del Conocimiento resulta en la actualidad una labor artesanal, propia para cada caso y dependiente de las personas concretas que estén involucradas en ella. Todo esto provoca que uno de los principales cuellos de botella en el desarrollo de un Sistema Basado en el Conocimiento sea el adquirir los conocimientos necesarios para poder construir sistemas eficientes. La información necesaria puede presentarse de múltiples formas, aunque conviene, en lo que respecta a los Sistemas Basados en el Conocimiento, considerar especialmente una serie de fuentes:

- **Libros y manuales.** Conocimientos básicos, específicos y públicos del dominio y del problema.
- **Documentación formal.** Documentos que contienen políticas, procedimientos, estándares, normas, regulaciones, leyes, etcétera, de un dominio. Este conocimiento también es de carácter público.
- **Documentación informal.** Notas, manuscritos, ayudas de trabajo, etc., proporcionan frecuentemente conocimiento heurístico para la resolución de problemas. Aunque a veces esta documentación es confidencial proporciona conocimientos semipúblicos.
- **Registros internos.** Registros de casos que se presentan, en forma de fichas de clientes, pacientes, estudios, estadísticas, etcétera. Pueden estar en forma escrita o, cada vez más, en forma digital (bases de datos). Además de para la validación y evaluación de los Sistemas Basados en el Conocimiento, esta información debe ser útil para la generación del propio conocimiento del sistema, uno de los objetivos de este curso.
- **Presentaciones.** Material usado para la formación, impartida o recibida. Tiene la ventaja de contener conocimientos expuestos de una forma muy clara.
- **Publicaciones.** Revistas especializadas, actas de congresos, etc.
- **Investigación.** Resultados de las investigaciones que se estén llevando a cabo, en forma de datos empíricos, informes, resultados estadísticos, etc.

- **Visitas.** El Ingeniero de Conocimiento se desplaza a los centros de trabajo del experto y los usuarios, para observar *in situ*.^{el} *modus operandi*.
- **Conocimiento humano** Además de las entrevistas con los expertos, resulta imprescindible la interacción con los directivos y usuarios. Los directivos pueden aportar objetivos del proyecto, alcance del sistema, contexto donde irá instalado, etcétera. Los usuarios deben dar claves de interfaces, necesidades, requisitos, etcétera.
- **WWW, audio, video ...**

Se suele denominar ‘Extracción de Conocimiento’ si la fuente es un documento y ‘Educción de Conocimiento’ si fuente es humana.

El Ingeniero de Conocimiento debe controlar constantemente qué información necesita, con qué profundidad, sobre qué temas, qué técnica debe utilizar para adquirir ese conocimiento y otros factores. Muchas veces resulta tentador improvisar, lo que frecuentemente provoca resultados negativos y falta de rigor. Por estas razones, el método que se presenta en este tema contempla básicamente tres grandes bloques:

1. Evaluación de viabilidad, definición del problema, primeras reuniones.
 2. Extracción de conocimientos de la documentación (incluidas bases de datos y demás fuentes documentales anteriormente mencionadas).
 3. Educción de conocimientos del experto, directivos y usuarios.
-
1. Comprensión general de la tarea y estructura funcional del Sistema Basado en el Conocimiento.
 2. Proceso de razonamiento del experto y pasos en la resolución de problemas.
 3. Datos necesarios para resolver un problema determinado, con los valores que pueden tomar.
 4. Desarrollo de un modelo conceptual.

Además de los métodos usuales de Adquisición de Conocimiento: Métodos automáticos (Machine Learning), a partir de ejemplos, inducción. Análisis estructural de textos. Entrevistas (abiertas, estructuradas). Observación de tareas habituales. Clasificación de conceptos. Cuestionarios. Análisis de protocolos. Emparrillados (Repertory Grids). Técnicas para educación en grupo. Método Delphi, se usa KDD y Minería de Datos para manipular grandes volúmenes de datos, con el fin de proporcionar información útil y conocimiento a partir de los mismos.

El Ingeniero de Conocimiento no suele coincidir con el experto. (A veces es deseable que el I.C. sepa algo del dominio para reducir tanto el tiempo como la paciencia que ha de tener el experto). El proceso de Adquisición de Conocimiento es difícil para el I.C. e incluso frustrante por la disponibilidad de tiempo del experto, la no existencia del experto adecuado y su resistencia a abandonar secretos. Las etapas que ha de cubrir el proceso de Adquisición de Conocimiento son las siguientes:

1. **Identificación del dominio y definición del problema** Están íntimamente relacionados ya que al identificar el dominio veremos el tipo de problemas que queremos resolver. Al considerar los problemas a resolver, identificaremos el dominio. Se trata de identificar el alcance de la aplicación del Sistema Basado en Conocimiento, así como la gama de problemas que pueden ser resueltos por él.
2. **Identificación de conceptos y relaciones del dominio** Ingeniero de conocimiento y experto han de cooperar intensamente para identificar conceptos, anotarlos y organizarlos. El conocimiento del experto se ha de formalizar de forma abstracta usando conceptos y relaciones (eliminar casuísticas particulares o anécdotas). Se ha de conseguir obtener un conocimiento organizado, coherente, inequívoco y no redundante. Pueden surgir problemas tales como: capacidad de expresión clara de conceptos, capacidad de separación de anécdotas y conceptos, de cosas fortuitas y sistemáticas o conocimiento incompleto.
3. **Formalización del conocimiento** El resultado de esta etapa será lo que posteriormente se representará y será el contenido de la base de conocimientos donde objetos y sus relaciones quedan reflejados. En esta etapa a partir de los conceptos identificados se han de inferir las relaciones entre ellos de forma que se consiga obtener un conocimiento organizado, coherente, inequívoco y no redundante. Esto está relacionado fuertemente con el proceso de representación del conocimiento. Razones: La naturaleza y estructura del conocimiento adquirido puede sugerir el método más apropiado de representación. Si el método de representar está prefijado hay que acomodar el conocimiento recogido a ese tipo de estructura. Problemas que ha de tener presentes el IC: Cuántos elementos de representación utilizará para formalizar el conocimiento y con qué detalle. Sistema de prioridades entre elementos. Interdependencia entre elementos. Orden de activación y disparo. Cómo añadir o borrar conocimiento. Una vez formulado, el conocimiento puede codificarse con algún esquema de representación.
4. **Chequeo** Requiere realizar una comprobación intensiva de la calidad y bondad del conocimiento recogido para corregir errores antes de usarlo. Método: revisión metódica y buenas baterías de casos ejemplo. Posibles fuentes de error: Conocimiento incompleto, inadecuada experiencia del experto, poca representatividad, hechos o relaciones olvidadas en la codificación, expresiones no válidas, falta de las heurísticas esenciales.

5. **Refinado y validación del conocimiento** Su objetivo es averiguar si el sistema ha alcanzado ya el suficiente grado de madurez. Se realiza con el sistema basado en conocimiento ya desarrollado. Se comprueba que resultan las mismas respuestas a cuestiones específicas que las que hubiese dado el experto para la solución de los mismos problemas planteados. Eliminación de expresiones inalcanzables, condiciones con hechos inalcanzables, redundancias y ciclos, inconsistencias, incompatibilidades y conflictos.

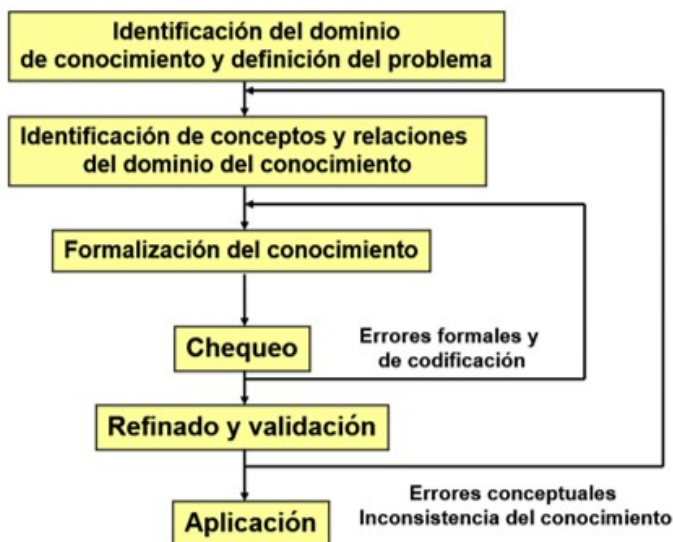


Figura 6.9: Etapas del proceso de adquisición de conocimiento.

La **ENTREVISTAS** son el método más comúnmente utilizado en la adquisición del conocimiento. Las entrevistas pueden ser estructuradas o no estructuradas, directas e indirectas y pueden contener cuestiones que son explícitas o implícitas. Son de tipo retrospectivo, es decir, se requiere que el experto recuerde situaciones pasadas. Las entrevistas han de registrarse (escritas, audio, video, etc.).

Las entrevistas son realizadas por el Ing. del Conocimiento al experto. El papel del I.C. consiste en: Cuidadosa preparación de entrevistas con vistas a recoger el conocimiento y a que el experto se involucre en la dinámica y conozca gradualmente cómo se va construyendo la base de conocimientos. Descomponer un problema en subproblemas discretos en la medida de lo posible. Hábil y prudente conducción de las entrevistas pasando gradualmente de cosas generales a más profundas. Contraste con otras fuentes de información recomendadas por el experto. Exacta y extensa documentación del conocimiento recogido y de los hechos en los que se ha puesto más énfasis. Validación y exacta interpretación de los argumentos del

experto por revisión de las entrevistas. Proposición de estructura de conocimiento con vistas a desarrollar un rápido prototipo de base de conocimientos. Especificar cómo deben ser las interacciones con el usuario durante las consultas. Con el paso del tiempo el número de sesiones con el experto va disminuyendo.

Se requiere del I.C. una serie de habilidades para llevar a cabo las entrevistas y la posterior representación del conocimiento: Comprender el lenguaje que usa el experto y comunicarse con él usando su vocabulario. Identificación de lo que es importante y separarlo de aquello más débil o menos importante, así como identificar aquellas partes donde pudiera haber huecos o lagunas. Integrar gradualmente en el conocimiento existente aquellas partes que tengan especial relevancia. Analizar el conocimiento adquirido y construir un modelo que refleje el mundo modelado. Seleccionar la estrategia de razonamiento más adecuada. Saber proponer el interfaz del SE apropiado.

Entrevistas no estructuradas o entrevistas abiertas

Son entrevistas espontáneas sin formato entre el experto y el Ing. del conocimiento. En ellas ni el contenido ni la secuencia de temas están predeterminados, aunque sí la orientación. Adecuadas en las primeras sesiones de Adquisición del Conocimiento para establecer el marco del dominio del conocimiento, conocer el lenguaje que usa el experto y tener una visión amplia del dominio. Se requiere una cierta habilidad para planificar la realización y conducción de este tipo de entrevistas sin que se vayan por “las ramas”. En cada sesión el IC habrá fijado un tema o perspectiva a tratar con el experto, así como un cierto grado de profundidad de los conocimientos a educir.

Típicas preguntas para comenzar una entrevista abierta: ¿Cómo resuelves este problema? ¿Cuáles son los elementos que influyen cuando resuelves el problema? ¿Cuáles son las informaciones que necesitas antes de empezar el tratamiento del problema? ¿Puedes describirme el último caso que has tratado? Una vez que el experto empiece a comentar un caso, se le debe interrumpir lo menos posible, aunque sí para preguntar algo que no se entiende

Es interesante alentar al experto con preguntas del estilo de: ¿Qué haces a continuación? (Esto ayuda al IC y anima al experto con el tema que desarrolla o salta a otro relacionado), ¿Puedes describir lo que quieres decir con eso? (Permite clarificar malos entendidos sobre la terminología), ¿Por qué haces esto? (La respuesta ayuda a revelar muchas de las conexiones y relaciones que usa el experto en esta parte de la decisión).

Entrevistas estructuradas

Son entrevistas entre el experto y el I.C. con un formato predeterminado. El grado de estructuración puede ser desde muy alto, donde el contenido y orden de los sucesos están predeterminados, a semi-estructurado, donde el contenido está fijado pero el secuenciamiento puede variar sobre la marcha. Las cuestiones a tratar pueden ser abiertas, imponiendo mínimas restricciones sobre la respuesta (cuestiones tipo qué, cómo, por qué...), o bien pueden ser cerradas, imponiendo restriccio-

nes importantes en la respuesta. Ventaja fundamental es que el hecho de tener una estructura permite un método de adquisición más sistemático y por tanto un mejor control sobre la cobertura del dominio de conocimiento. Dificultad: requiere mayor tiempo de preparación y algún conocimiento previo.

Las preguntas a plantear en una entrevista estructurada se deben centrar sobre los conocimientos de conceptos, relaciones e inferencias del experto. Típicas preguntas de una entrevista estructurada son las siguientes: ¿Qué tipo de cosas te gustaría saber acerca del problema cuando empiezas a sopesarlo? ¿Este valor depende de otros factores? En caso afirmativo ¿cuáles son? ¿Es esto lo que haces en esta situación? ¿Están incluidos aquí todos los conceptos relacionados con tal tema? ¿Te parece que este diagrama muestra correctamente el orden de tus decisiones?

Variantes de entrevistas estructuradas

- **Discusión focalizada:** Centrada en tipos particulares de información (casos, objetivos, diagramas de información, etc.).
- **Análisis de casos tipo:** Centrada en casos vividos por el experto. Consiste en hacer recordar al experto cómo se desarrolló el caso (incluso si es posible simulando la actuación). Particularmente relevante para procesos en tiempo real.
- **Centrada en casos vividos por el experto.** Consiste en hacer recordar al experto cómo se desarrolló el caso (incluso si es posible simulando la actuación). Particularmente relevante para procesos en tiempo real.
- **Simulación de escenarios hacia delante:** Se estudia un caso por simulación. El IC propone unas hipótesis de partida al experto y éste trata de simular su actuación. Se requiere gran conocimiento por parte del IC para proponer los casos.
- **Método de los incidentes o decisiones críticas:** Se basa en casos importantes. Se debería tener presente Un cronograma de la secuencia de hechos o acciones tomadas en el caso.
- **Método de los objetivos:** Se basa en que el experto proponga una lista con pares de objetivos que sepa separar. Después de construida la lista se tendrán los rasgos mínimos que diferencian todos los objetivos.
- **Método de la división del dominio:** Se basa en que el experto llegue a conclusiones a partir de síntomas o hechos (o su agrupamiento).
- **Método de la reclasificación o descomposición de metas:** Se basa en que el experto a partir de una meta vaya descomponiéndola en sub-metas o tareas hasta llegar a un conjunto de síntomas (trabajo del experto hacia atrás).
- **Método del “Teachback”:** El experto explica algo al IC y después éste explica al experto lo mismo. Este proceso continúa hasta que el experto lo considere oportuno.

- **Método del “Role” inverso:** El IC y el experto invierten los papeles. Se diferencia del “teachback” en que hay actuación.
- **Método de las veinte preguntas:** El IC se piensa una situación y el experto ha de adivinar mediante preguntas, qué piensa el IC. (Contestaciones si/no). Se requiere conocimiento por parte del IC. Este problema se puede mitigar usando un segundo experto en el puesto del IC.
- **Método del escenario incompleto:** El IC presenta un escenario al experto en que sabe que falta información relevante. Se trata de que el experto identifique lo omitido.

Técnicas de entrevista en grupo

Consiste en la adquisición del conocimiento de entrevistas entre varios expertos al mismo tiempo junto con el IC. Problema de la discrepancia de los puntos de vista que hay que consensuar (pero puede no corresponder con la actuación de un experto). Ventaja de que unos expertos pueden incitar a otros a expresar conocimientos que el IC no llegaría a provocar.

- **“Brainstorming”:** Énfasis en el número de respuestas.
- **Toma de decisiones por consenso:** Enfatiza la calidad de las respuestas.
- **Método Delphi:** Cada experto examina las respuestas de otros expertos y las reevalúa iterando con ellos. El proceso continua hasta llegar a un consenso.
- **Técnica del grupo nominal.** (Similar al método Crawford Slip). Los expertos contestan en papel a una serie de cuestiones. Estas respuestas son puestas bajo un formato común de donde se extraen los conceptos principales (Se puede iterar con la información formateada).

6.3.4. Etapa 4: Representación del conocimiento

Conceptualización.

Realizada la Adquisición del Conocimiento, es necesaria una etapa intermedia antes de representar el conocimiento adquirido. Esta fase se suele denominar **CONCEPTUALIZACIÓN** y consiste en (según Buchanan): “hacer explícitos los conceptos claves y las relaciones relevantes”. La conceptualización establece qué conocimientos maneja el experto, cómo los utiliza, dónde los emplea y cuándo los usa. La conceptualización permite obtener una DESCRIPCIÓN DETALLADA DEL PROBLEMA a resolver dando lugar a modelos conceptuales

Durante la conceptualización se deben determinar los 3 tipos de conocimientos siguientes:

- **Estratégicos o de control:** Especifican qué hacer, dónde y por qué hacerlo, es decir, fijan la secuencia de pasos que el SE deberá seguir para ejecutar su tarea.

- **Tácticos:** de acción u operativos, especifican cómo y cuándo el SE puede añadir a sus conocimientos genéricos información actual acerca del caso.
- **Factuales o declarativos:** especifican lo que es, o se cree que es verdad acerca del mundo en general y del caso particular para el cual se está ejecutando la tarea.

La conceptualización es un par $M=C, R$ que incluye los Conceptos que van a existir y sus Relaciones.

Concepto: Puede ser cualquier cosa acerca de la cual se quiere decir algo, se usa en un sentido bastante amplio:

- Cosas **concretas** (objetos, personas...).
- Cosas **abstractas** (el número 2, el conj. de los enteros...).
- Conceptos **elementales** (electrón...).
- Conceptos **compuestos** (átomo...).
- Conceptos de **ficción** (unicornio...).
- Conceptos **reales** (detective...).
- Descripción de **tareas, acciones, estrategias...**

Relación: Es un tipo de interacción entre conceptos en un universo de discurso. Tienen las siguientes propiedades:

- **Valencia:** Describe la forma cuantitativa en la que los elementos intervienen en la relación (uno-a-uno, muchos-a-uno...).
- **Funcionalidad:** Señala el tipo de implantación de la relación (inyectiva o monomórfica, sobreyectiva o epimórfica...).
- **Cardinalidad:** Número de argumentos que intervienen en la relación (unaria, binaria...).

Las más típicas son:

- **Taxonómica.** Clasifica un concepto específico en otro general.
 - A es B
 - A puede clasificarse como un B, un C o un D
- **Causal.** Describe cómo ciertos estados o acciones inducen otros estados o acciones.
 - A causa B
 - A es causado por B
 - A es efecto de B

- Siempre que A, B
- **Estructurante.** Describe cómo un concepto o un sistema de conceptos puede ser descompuesto en partes o subsistemas.
 - A es parte de B
 - A está formada por B y C
 - A forma parte de B
 - A es disjunto de B
 - A es no disjunto de B
 - A y B son particiones de C
 - A y B son particiones exhaustivas de C
- **Topológica.** Describe la distribución espacial de conceptos físicos y las interconexiones entre esos conceptos.
 - A está a la derecha de B
 - A está encima de B
 - A está dentro de B
 - A contiene a B
 - A intersecta a B
 - A está conectada a B
 - A está en contacto con B
- **Funcional.** Describe las condiciones por las cuales pueden ocurrir acciones y las reacciones y consecuencias que pueden resultar de las acciones.
 - A permite B
 - A necesita B
 - A dispara B
- **Similitud.** Establece la “igualdad” o analogía entre conceptos y en qué grado.
 - A igual a B
 - La válvula A esta abierta = admite gasolina
- **Cronológica.** Describe la secuencia temporal en la que ocurren los eventos.
 - A sucede antes de B
 - A ocurre después de B
 - A y B suceden simultáneamente
 - A ocurre durante B
 - A comienza antes de que finalice B

- **Condicional.** Define las condiciones en que tienen lugar ciertas cosas.
 - Cuando la velocidad es la de la luz y el espacio infinito, usar hipótesis relativista.
- **Finalidad.** Establece el por qué y el para qué de los conceptos.
 - El airbag fue diseñado para evitar consecuencias graves en los accidentes.

La identificación de conceptos o subconjuntos funcionales del más alto nivel, así como la terminología clave, se describen en el Diccionario de Conceptos. Suele construirse como una **tabla Objeto-Atributo-Valor** (Figura 6.10).

OBJETO	ATRIBUTO	VALOR
PERSONA	Nombre	{Juan, Ana, Pedro}
	Edad	[0, 120] años.
	dirección	<calle, nº, CP>

Figura 6.10: Ejemplo de tabla Objeto-Atributo-Valor.

El **MAPA DE CONOCIMIENTOS** debe reflejar completamente el proceso de razonamiento del experto. Puede ser visto como una síntesis de los dos elementos anteriores (objetos-diccionario de conceptos y tareas-modelo dinámico de procesos). No existe consenso en la forma exacta de representarlo. Suele ser propio de cada IC y de los dominios diferentes. Se suele representar con esquemas (similares a los “diagramas de flujo”).

Algunos consejos a la hora de construir el Mapa de Conocimientos:

- Identificar claramente las decisiones a ser realizadas por el SE.
- Usar los resultados de las representaciones externas intermedias.
- Evitar duplicar inferencias.
- Examinar condiciones desconocidas y por defecto.
- Confrontar la imprecisión y la incertidumbre.

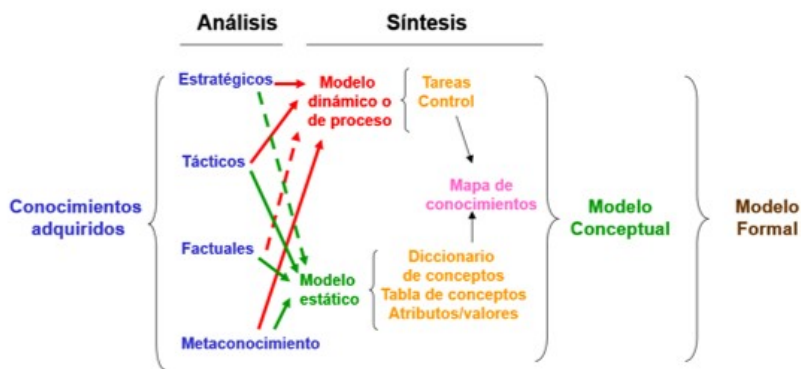


Figura 6.11: Resumen del proceso de conceptualización.

Representación del conocimiento.

La representación del conocimiento suele ser una de las tareas que más tiempo y esfuerzo demandan de las personas que se dedican al diseño de SBCs. Ello se debe al hecho de no conocer aún con la suficiente precisión cuáles son los procesos que cualquier ser humano y, en particular un experto, activa en su mente cuando selecciona, examina, sintetiza y transforma los datos iniciales de un problema para alcanzar una solución válida.

La función de cualquier esquema de representación es capturar los rasgos esenciales del ámbito correspondiente a un problema concreto y hacer accesible esta información a un procedimiento de resolución específico. Como se sabe, diferentes tipos de problemas requieren diferentes tipos de razonamiento. A su vez, cada modalidad de razonamiento precisa de una adecuada representación del conocimiento.

Se pretende hacer una correspondencia entre el dominio de nuestra aplicación y un sistema de símbolos que usará nuestro SE. Hay que identificar qué conocimiento hay que codificar y cómo codificarlo para su posterior uso.

La representación del conocimiento que habrá en la base de conocimientos de nuestro SE ha de ser adecuada al conocimiento adquirido, ha de representar todos los tipos de conocimiento de nuestra aplicación y ser adecuada para inferir.

La Lógica Formal es la forma más antigua conocida de representar conocimiento y aceptada por la comunidad científica. Se ha utilizado por ser clara, transparente, no ambigua y simple a la hora de representar hechos y sus relaciones y fácil de desarrollaren software. La base de conocimientos consta de expresiones en lógica formal, fácilmente analizables.

Las “reglas de producción” son la forma más extendida de representar el conocimiento, constan de Condiciones (Hipótesis) y Acciones (Conclusiones) y tienen la forma:

Si (If)
Condición 1

```

y Condición 2
.....
y Condición n
Entonces (Then)
Conclusión 1
y Conclusión 2
.....
y Conclusión m

```

Es la forma más extendida de representar el conocimiento. Ejemplo de Regla de producción:

```

Si
ha fallado la bombilla
y hay una de repuesto
y está útil

Entonces
cambiar la bombilla por la de repuesto
y seguir trabajando

```

Los **“sistemas de producción”** son modelos de cálculo que han probado su eficiencia en la Ingeniería del Conocimiento tanto en el desarrollo de algoritmos de búsqueda como en el modelado de problemas del dominio humano. Su origen se remonta a 1943 a las ideas de Post. Consta de:

- Conjunto de reglas de producción.
- Memoria de trabajo (Memoria temporal).
- Ciclo de reconocer y actuar.

Memoria de trabajo: Contiene una descripción del estado actual del mundo o entorno de la aplicación en cada paso del proceso de razonamiento. Esta descripción es un modelo que servirá para asociar las partes condición de las reglas con las observaciones del mundo con objeto de seleccionar o producir las apropiadas acciones. En el momento en que se cumplen todas las condiciones de una regla se produce el “disparo” de la misma ejecutándose la acción. Esta operación alterará el contenido de la memoria de trabajo.

Ciclo de reconocimiento y actuación: Es el procedimiento de control de un sistema de producción. Es un procedimiento de marcha hacia adelante. La memoria de trabajo se inicializa con la descripción del problema. Los modelos guardados en la memoria de trabajo se tratan de superponer en las condiciones de las producciones. Se crea un conjunto “conflicto”. Este es un subconjunto de producciones cuyas condiciones se cumplen. Se escoge una producción y se “dispara” o se activa. La

acción de la regla es disparada cambiando el contenido de la memoria de trabajo. Se repite todo el proceso descrito con la memoria de trabajo modificada. El proceso continua hasta que no hay condiciones en las reglas que cumplan el contenido de los modelos de la memoria de trabajo.

La utilización de los sistemas de producción en SE se remonta a los trabajos de Newell y Simon de la Universidad Carnegie Mellon entre 1960 y 1970. Vieron que los sistemas de producción podían utilizarse para modelar el conocimiento humano y hacer un posterior uso del mismo en la inferencia.

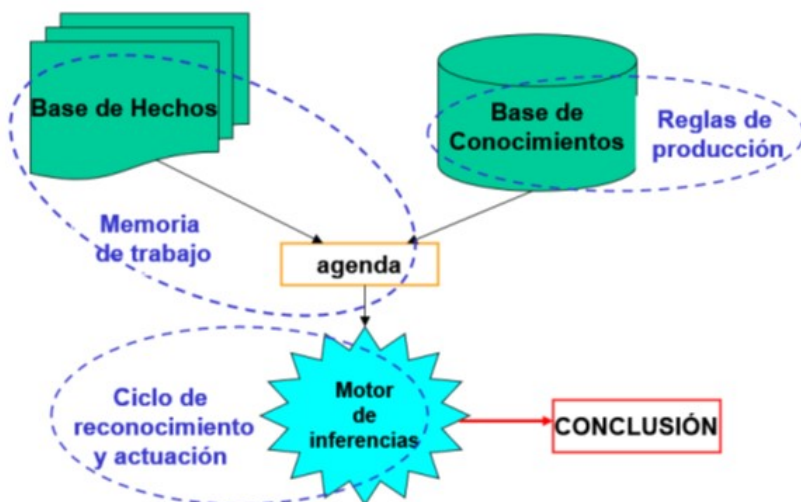


Figura 6.12: Sistemas de producción y sistemas expertos.

Las principales ventajas de los sistemas de producción en los S.E son la separación del conocimiento y del control. Se pueden hacer cambios fáciles de reglas sin cambiar el control y viceversa. Modularidad de las reglas de producción e independencia del lenguaje de programación usado.

Razonamiento.

El razonamiento se puede definir como el proceso de obtención de inferencias o conclusiones a partir de unos hechos u observaciones reales o asumidos y de un conocimiento previo. La inferencia es el proceso por el que a partir de unos hechos conocidos se obtienen conclusiones acerca de otros desconocidos.

El razonamiento automático ya se usaba en los 50 en juegos. En 1963 se presentó el sistema “General Problem Solver” capaz de hacer inferencias lógicas (Newell y Simon). Aparición del razonamiento heurístico.

Tipos de razonamiento en SE:

- **Forward chaining** (encadenamiento hacia delante, deductivo, progresivo, dirigido por datos o hechos) Síntomas ->Causas

- **Backward chaining** (encadenamiento hacia atrás, inductivo, regresivo, dirigido por metas u objetivos) Síntomas <- Causas
- Forward/Backward chaining.

Pasos del motor de inferencia:

1. Elaboración de un conjunto conflicto con todas las reglas cuyas condiciones se cumplen.
2. Detección (filtro) de reglas pertinentes o selección de reglas a partir de unos hechos. Se trata de obtener de la BC el conjunto de reglas aplicables en una situación determinada o estado de la BH.
3. Aplicación de reglas o resolución del conflicto. Consiste en seleccionar una regla del conjunto conflicto y dispararla (ejecutar su conclusión). Se altera la BH o memoria de trabajo.
4. Vuelta a 1 hasta que el conjunto conflicto esté vacío.

Ciclo de “razonamiento hacia delante”:

1. Parte de unas observaciones (hechos).
2. A partir de los hechos observados se seleccionan las reglas cuyas condiciones están relacionadas con ellos.
3. Las reglas seleccionadas son examinadas para ver si verifican todas sus condiciones. Aquéllas que las verifican constituyen el conjunto conflicto“.
4. De todas aquellas que forman el conjunto conflicto se selecciona una sola y se activa (se dispara). La selección de una regla del conjunto conflicto es la resolución del conflicto”
5. La activación de la regla provocará la aparición de otros hechos que se añaden a los observados y se actualiza la base de hechos.
6. Volver al paso 2 hasta analizar todos los hechos observados y deducidos.

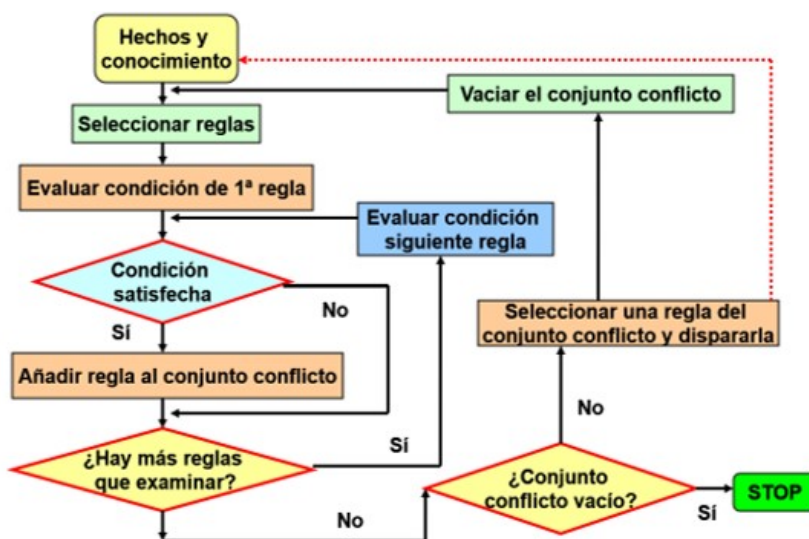


Figura 6.13: Ciclo de razonamiento hacia delante.

6.3.5. Etapa 5: Implementación

En esta fase se lleva a cabo el desarrollo íntegro del SBC en términos de programación. Se suelen construir prototipos que serán confrontados con el experto y rediseñados en numerosas ocasiones. Así se sigue una metodología incremental y cíclica con la que el sistema se va refinando hasta conseguir optimizarlo. También es importante decidir el lenguaje de programación utilizado para la implementación dependiendo de factores como los siguientes: requerimientos de tiempos de respuesta, requerimientos de Interfaz de usuario, flexibilidad que se quiere aportar a la herramienta, requerimientos de hardware, coste en mantenimiento de la herramienta y facilidades aportadas al usuario.

6.3.6. Evaluación del SE.

La evaluación de un Sistema Basado en el Conocimiento no es una tarea sencilla, debido, entre otras, a las siguientes razones:

- Las expectativas con respecto a sus prestaciones son frecuentemente vagas y confusas.
- Muchas veces los casos típicos que debe resolver el sistema son los mismos que se han usado para la adquisición de conocimiento.
- No es posible en un caso real disponer de un juego de pruebas que pueda asegurar que el comportamiento del sistema es el ideal.

Además, comparándolo con la evaluación de un software convencional, la evaluación de un SBC posee las siguientes características diferenciales:

- Los SBC no son por naturaleza objetivos.
- Los SBC gestionan habitualmente cierta cantidad de incertidumbre.
- En el software convencional puede usarse el juego de ensayo.
- En el software convencional normalmente no se cuestiona la corrección de los resultados.

La verificación del sistema consiste en comprobar la corrección de los modelos conceptuales, formales y computables del sistema. La debe realizar un Ingeniero de Conocimiento distinto al que desarrolló el sistema:

- **Redundancias:** Identidad entre elementos. Elementos sin conocimiento. Circularidades. Partes innecesarias.
- **Complejidad:** Elementos ausentes. Elementos inalcanzables.
- **Consistencia:** (Depende de cada formalismo de representación utilizado, de los tipos de lógicas y de los mecanismos de gestión de incertidumbre): Elementos contradictorios. Incompatibilidades con las restricciones.

La validación consiste en comprobar que el sistema responde adecuadamente a los fines para que fue diseñado. La debe realizar el experto, o utilizarse conjuntos de casos de prueba, pero en general, se deben realizar los siguientes procesos (Figura 6.14):

		Carga de Prueba	Patrón de referencia
Comparación frente a resultados conocidos		Casos históricos con solución conocida	Comparar resultado del sistema con el conocido
Comparación frente a modelos matemáticos	Modelo Teórico	Casos aleatorios	Comparar resultado del sistema con la solución del modelo para el caso
	Modelo lineal		
Comparación frente a un experto	Juego aleatorio	Casos aleatorios	Comparar resultado del sistema con el del experto
	Juego de ensayo	Casos seleccionados	
	Ensayo en paralelo	Casos reales	
Análisis de sensibilidad	Sensibilidad a variaciones en las entradas	Casos aleatorios donde se provocan variaciones en la entrada	Las salidas de los casos variados deben ser semejantes
	Sensibilidad a errores en las entradas	Casos con entradas inválidas	No debe producirse salida y debe dar mensaje de error

Figura 6.14: Validación de un SE.

También deben ser evaluadas la usabilidad y la utilidad del sistema, cuando el sistema está funcionando en condiciones reales.

6.4. Bibliografía

1. Gómez, A., Juristo, N., Montes, C, Pazos, J. (1997). Ingeniería del conocimiento. Ed. Centro de estudios Ramón Areces.