CAPITULO 1

VERIFICACIÓN Y VALIDACIÓN DE SISTEMAS INTELIGENTES

- **Verificación de Sistemas**
- **Validación de Sistemas**
- **Métodos Cuantitativos de Validación**
- **Metodología de Validación**
- **Herramientas de Validación**
- **Resumen**
- **Textos Básicos**

1. VERIFICACIÓN Y VALIDACIÓN DE SISTEMAS INTELIGENTES

Verificación y validación (V&V) son dos de las etapas más importantes en el análisis del comportamiento de un sistema experto. Sin entrar en grandes profundidades, con la verificación trataremos de comprobar si hemos construido nuestro sistema correctamente. Ello implica asegurarse de que el "software" implementado no contiene errores, y que el producto final satisface los requisitos y las especificaciones de diseño. Por otra parte, el término validación se refiere, más bien, a un análisis de la calidad del sistema inteligente en su entorno real de trabajo, lo que nos permite determinar si el producto desarrollado satisface convenientemente las expectativas inicialmente depositadas.

Ambas fases, verificación y validación, forman la base de un entramado más complejo destinado a evaluar globalmente el comportamiento de un sistema inteligente. Por simplicidad, las fases posteriores a la V&V se agrupan bajo el termino evaluación.

La evaluación se encarga de analizar aspectos que van más allá de la corrección de las soluciones finales del sistema. Así analizaría aspectos como utilidad, robustez, velocidad, eficiencia, posibilidades de ampliación, facilidad de manejo, análisis coste/beneficio, etc.

Todas estas fases se organizan jerárquicamente como se ilustra en la Figura 1.1.

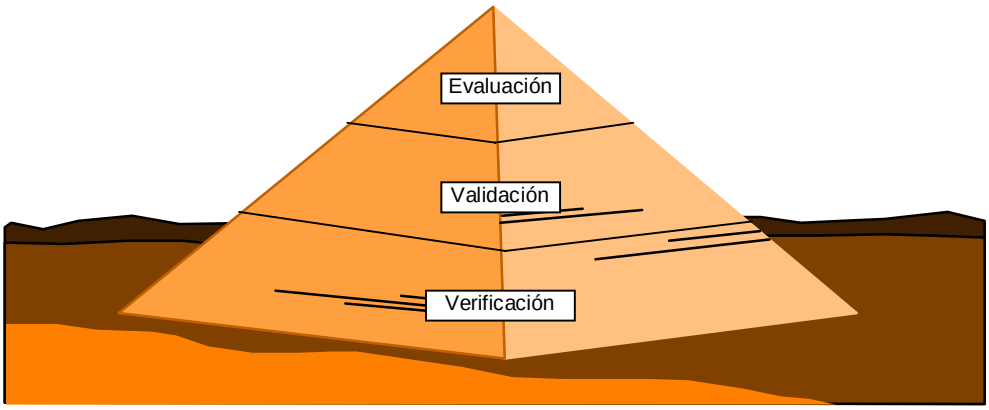


Figura 1.1 Pirámide del análisis del comportamiento de un sistema inteligente.

1.1. Verificación de Sistemas Inteligentes

La verificación de sistemas expertos es un proceso que incluye las siguientes tareas: (a) verificación del cumplimiento de las especificaciones, (b) verificación de los mecanismos de inferencia, y (c) verificación de la base de conocimientos.

1.1.1 Verificación del cumplimiento de las especificaciones

El análisis del cumplimiento de las especificaciones puede ser llevado a cabo por los desarrolladores, los usuarios, los expertos y/o un grupo de evaluadores independientes. En el software convencional este proceso está cada vez más automatizado con el advenimiento de las herramientas de ingeniería del software asistida por ordenador (CASE). Sin embargo, la inclusión de estas herramientas en el ámbito de la ingeniería del conocimiento es lenta. Las cuestiones a analizar en este proceso consisten en comprobar si:

- Se ha implementado el paradigma de representación del conocimiento adecuado.
- Se ha empleado la técnica de razonamiento adecuada.
- El diseño y la implementación han sido llevados a cabo modularmente.
- La conexión con el software externo se realiza de forma adecuada.
- El interfaz de usuario cumple las especificaciones.
- Las facilidades de explicación son apropiadas para los potenciales usuarios del sistema.
- Se cumplen los requisitos de rendimiento en tiempo real.
- El mantenimiento del sistema es posible hasta el grado especificado.
- El sistema cumple las especificaciones de seguridad.
- La base de conocimientos está protegida ante modificaciones realizadas por personal no autorizado.

1.1.2 Verificación de los mecanismos de inferencia

El uso de shells comerciales ha reducido la dificultad de la verificación de los mecanismos de inferencia, ya que se asume que ésta ha sido realizada por los desarrolladores de la herramienta. La responsabilidad del ingeniero del conocimiento recae fundamental-mente en la elección de la herramienta apropiada.

Sin embargo, esta asunción de correcto funcionamiento no siempre es cierta (sobre todo en versiones nuevas de la herramienta). Por ello, para aplicaciones que trabajan en dominios críticos, el funcionamiento correcto debe verificarse a través de distintas pruebas.

Muchas veces los problemas con las shells comerciales pueden no ser la causa de errores en su programación. Así, en ocasiones hay que pensar en un desconocimiento del funcionamiento exacto de la herramienta. Por ejemplo, los procedimientos de resolución de conflictos o los mecanismos de herencia pueden hacer difícil el

seguimiento del curso exacto de la inferencia. De esta forma, aunque el conocimiento estático esté verificado, el funcionamiento final del sistema puede no ser el apropiado.

En caso de que decidamos construir nuestros propios mecanismos de inferencia, será preciso realizar su verificación. Como estamos tratando software convencional podemos aplicar para su verificación las técnicas diseñadas para la verificación dentro de la ingeniería del software.

Generalmente se recomienda la utilización, siempre que sea posible, de mecanismos de inferencia certificados cuyo funcionamiento correcto se haya probado. Además en caso de utilizar herramientas comerciales aconsejan realizar pruebas para comprobar que realmente se comportan como indican en sus manuales.

1.1.3 Verificación de la base de conocimientos

La verificación de la base de conocimientos, a diferencia de los mecanismos de inferencia, es plena responsabilidad del ingeniero del conocimiento. Esta verificación se basa, generalmente, en el concepto de *anomalías*. Una anomalía es un uso poco común del esquema de representación del conocimiento, que puede ser considerado como un error potencial (existen anomalías que no constituyen errores y viceversa).

La verificación de la base de conocimientos no nos asegura que las respuestas de nuestro sistema sean correctas, lo que nos asegura es que el sistema ha sido diseñado e implementado de forma correcta.

La mayoría de los estudios publicados que tratan sobre la verificación de las bases de conocimientos se refieren a los sistemas basados en reglas, ya que son los más populares. Por ello en este capítulo nos centraremos en dichos sistemas. Esto no quiere decir que los sistemas construidos según otros paradigmas no necesiten ser verificados o que su verificación no sea posible. Así, por ejemplo, Cheng (1989) muestra como se llevaría a cabo la verificación de un sistema inteligente basado en frames; Shiu et al. (1997) realizan una verificación formal de un sistema que utiliza reglas y frames; y Kandelin y O´Leary (1995) realizan la verificación de un sistema orientado a objetos.

Aspectos que se suelen examinar a la hora de verificar una base de conocimientos son la consistencia y la completitud. En la Tabla 1.1 vemos una serie de pruebas que se realizan para comprobar que la base de conocimientos es consistente y completa. En principio supondremos que los sistemas no manejan incertidumbre, luego veremos como la inclusión de incertidumbre puede afectar a las pruebas desarrolladas.

Consistencia	Reglas redundantes	$p(x) \wedge q(x) \rightarrow r(x)$ $q(x) \wedge p(x) \rightarrow r(x)$
	Reglas conflictivas	$p(x) \wedge q(x) \rightarrow r(x)$ $p(x) \wedge q(x) \rightarrow \neg r(x)$
	Reglas englobadas en otras	$p(x) \wedge q(x) \rightarrow r(x)$ $p(x) \rightarrow r(x)$
	Reglas circulares	$p(x) \rightarrow q(x)$ $q(x) \rightarrow r(x)$ $r(x) \rightarrow p(x)$
	Condiciones IF innecesarias	$p(x) \wedge q(x) \rightarrow r(x)$ $p(x) \wedge \neg q(x) \rightarrow r(x)$
Completitud	Valores no referenciados de atributos	Ocurre cuando algunos valores, del conjunto de posibles valores de un atributo, no son cubiertos por la premisa de ninguna regla.
	Valores ilegales de atributos	Una regla referencia valores de atributos que no están incluidos en el conjunto de valores válidos para ese atributo.
	Reglas inalcanzables	$p(x) \rightarrow r(x)$ p(x) no aparece como conclusión de otra regla ni puede obtenerse del exterior (raz. progresivo)
	Reglas sin salida	$p(x) \wedge q(x) \rightarrow r(x)$ r(x) no es una conclusión final y no aparece en la premisa de ninguna regla (raz. progresivo)

Tabla 1.1 Verificación de la consistencia y la completitud en bases de conocimientos.

Influencia de las medidas de incertidumbre

Las reglas vistas hasta ahora para verificar la consistencia y la completitud son válidas siempre y cuando los sistemas no incluyan incertidumbre. En caso de que si exista dicha incertidumbre la validez de las pruebas queda en entredicho, ya que como veremos situaciones normales pueden ser tomadas como errores.

En sistemas que pretenden medir incertidumbres o grados de asociación (utilizando factores de certeza, probabilidades bayesianas o cualquier otro método) es importante verificar que estos valores son consistentes, completos, correctos y no redundantes. Esta tarea se realiza, en primer lugar, asegurándonos que cada regla incluye un factor de incertidumbre, y que estos factores cumplen los aspectos de la teoría en la que se basan.

La búsqueda de anomalías en las medidas de incertidumbre de un sistema inteligente es un proceso que no ha recibido mucha atención por parte de los investigadores, quizá debido al limitado número de sistemas expertos que hacen un uso extensivo de dichas medidas. Un ejemplo de verificación de este tipo es el trabajo que O'Leary (1990) desarrolló en sistemas que seguían el esquema bayesiano.

El modo en que el uso de medidas de incertidumbre también puede afectar a la realización de los tests de consistencia y completitud puede verse en los siguientes ejemplos (Nguyen et al., 1987):

- *Redundancia*: Si antes la redundancia no afectaba a la salida del sistema ahora puede causar graves problema ya que, al contar la misma información dos veces, su pueden modificar los pesos de las conclusiones.

- *Reglas englobadas en otras*: Esta situación puede no ser errónea ya que las dos reglas pueden indicar la misma conclusión pero con distintas confianzas. La regla englobada sería un refinamiento de la regla más general para el caso de que tengamos más información.

- *Condiciones IF innecesarias*: Igual que en el caso anterior, una condición IF innecesaria puede utilizarse para variar la confianza en la conclusión final.

- *Reglas circulares*: Pueden existir casos en los que la utilización de medidas de incertidumbre rompan la circularidad de un conjunto de reglas. Por ejemplo, si el factor de certidumbre de una conclusión implicada en el ciclo cae por debajo de un umbral (normalmente entre –0.2 y 0.2) se considera que el valor de la conclusión es "desconocido" y el ciclo se rompe.

- *Reglas "sin salida"*: La detección de este tipo de reglas se complica con la introducción de incertidumbre. Así, una regla puede convertirse en una regla "sin salida" si su conclusión tiene una certidumbre por debajo del umbral en el cual un valor se considera "conocido". Por ejemplo, la siguiente cadena de reglas

$$A \xrightarrow[0.4]{R1} B \xrightarrow[0.7]{R2} C \xrightarrow[0.7]{R3} D$$

  podría parecer válida, sin embargo si A se conoce con total certidumbre, el factor de certeza de D después de un razonamiento progresivo sería $0.4 \times 0.7 \times 0.7 = 0.196$ (menor que 0.2) con lo que el valor de D sería "desconocido" y la línea de razonamiento acabaría en un punto "sin salida".

- *Reglas inalcanzables*: de forma similar al ejemplo anterior pueden existir reglas que, por causa de los factores de certeza, se convierten en inalcanzables. Si consideramos el siguiente conjunto de reglas

$$A \xrightarrow[0.1]{R1} B \xrightarrow[1]{R2} C$$

  la regla R2 sería inalcanzable en un razonamiento progresivo (aunque su premisa aparece en la conclusión de otra regla) porque el valor de B cae por debajo del umbral de 0.2.

Verificación dependiente o independiente del dominio

La verificación de un sistema inteligente puede enfocarse desde dos puntos de vista diferentes: verificación dependiente del dominio, y verificación independiente del dominio.

La verificación independiente del dominio se basa en la detección de las anomalías a través de técnicas heurísticas mediante las cuales se analiza la base de conocimientos pero sin tener en consideración el dominio de aplicación.

Por el contrario, la verificación dependiente del dominio utiliza metaconocimiento del propio universo de discurso para examinar las bases de conocimiento implementadas. El ejemplo más conocido de este tipo de verificación lo encontramos en TEIRESIAS, que es un sistema que supervisa la introducción de conocimiento nuevo en el sistema experto MYCIN. Muchas herramientas actuales emplean mecanismos automáticos similares al de TEIRESIAS para impedir la introducción de conocimiento erróneo en un sistema en desarrollo. De todas formas, este procedimiento presenta algunas desventajas. Así, el metaconocimiento – que no es más que conocimiento sobre conocimiento –, también debe ser verificado. Además, el propio metaconocimiento puede no ser estable, si existe una aportación continua de conocimiento nuevo. Por último, el desarrollo de una aplicación que permita realizar verificaciones dependientes del dominio suele ser una tarea lenta y costosa, en parte por el hecho de tener que adquirir el metaconocimiento necesario, y en parte por el hecho de tener que mantenerlo.

Automatización de los mecanismos de verificación

De las distintas fases que componen el análisis del comportamiento de un sistema inteligente, la fase de verificación es en la que se ha conseguido un mayor grado de automatización mediante distintos tipos de herramientas. Dentro de estas herramientas de verificación podemos establecer dos grupos: las dependientes y las independientes del dominio. En la Tabla 1.2 se muestra un resumen de las principales características de las herramientas de verificación más nombradas en la bibliografía. Como vemos las herramientas dependientes del dominio hacen uso del metaconocimiento mientras que las independientes del dominio se basan, principalmente, en convertir la base de conocimientos en una representación independiente (mediante tablas o grafos) a partir de la cual se buscan las posibles anomalías.

Tipo	Nombre	Referencia	Características
Dependientes del dominio	TEIRESIAS	Davies (1976)	• Identifica errores en la base de conocimientos y los corrige mediante la modificación, adición o borrado de reglas • Adecuada para el desarrollo incremental del sistema
	EVA	Stachowitz y Combs (1987)	• Chequea las reglas de la base de conocimientos empleando metaconocimiento previamente desarrollado • Interactúa con shells como KEE trasladando las reglas a un formato propio
Independientes del dominio	RCP	Suwa et al. (1982)	• Permite realizar la verificación a medida que el sistema inteligente se va desarrollando • Analiza la base de conocimientos a partir de una tabla de decisión en la que se muestran todas las posibles combinaciones de valores que pueden tomar los atributos de condición y los relaciona con los correspondientes valores que concluiría el sistema.
	ESC	Cragun y Steudel (1987)	• También utiliza tablas de decisión
	CHECK	Nguyen et al. (1987)	• Extensión del trabajo de Suwa et al. • Incluye muchos criterios para la validación de las reglas y realiza tablas y gráficos de dependencias para mostrar las interrelaciones entre las distintas reglas
	COVER	Preece et al. (1992)	• Esta herramienta construye, directamente de las reglas, un grafo que las representa. La ventaja de esta técnica es que permite detectar anomalías entre numerosas reglas, y no sólo entre pares de reglas como es común en las aproximaciones basadas en tablas.
	KB-Reducer	Ginsberg (1988)	• Transforma las reglas en una representación basada en la lógica que permite detectar las anomalías
	Basadas en redes de Petri	Agarwal y Tanniru (1992) Wu y Lee (1997) etc.	• Se basan en el uso de redes de Petri lo que permite analizar las relaciones temporales entre las reglas. • Tiene el inconveniente de que la conversión de una base de reglas a una red de Petri no es una tarea trivial
	Validator	Kang y Bahill (1990)	• Emplea medidas estadísticas para encontrar errores en la base de conocimientos después de la ejecución de una serie de casos de prueba.

Tabla 1.2 Principales herramientas de verificación de sistemas expertos

No obstante, el principal problema que se le achaca a las herramientas de verificación de bases de conocimientos, es que suponen una serie de condiciones muy simplificadas para que su funcionamiento sea efectivo, como por ejemplo:

- *Tamaño reducido*: Los métodos descritos por las distintas herramientas suelen ser adecuados cuando el número de reglas no es muy elevado. Al aumentar el número de reglas puede producirse una explosión combinatoria que reduciría drásticamente la eficiencia de la herramienta. Ginsberg destacó que su herramienta, KB-Reducer, pasaba de tardar 40 segundos con una base de conocimientos de 50 reglas a tardar 10 horas con otra de 370 reglas. Para evitar este problema se han desarrollado técnicas como la partición de la base de reglas, como se hizo en ESC, o el uso de heurísticas, como en la herramienta COVER.

- *Bases de reglas no estructuradas*: La mayoría de las herramientas de verificación suponen que la base de reglas es *plana* y que la ejecución de las reglas se hace mediante una estrategia progresiva o regresiva pero sin tener en cuenta otras estructuras de control. Sin embargo, los dominios complejos y poco estructurados en los que se pretenden desarrollar sistemas expertos suelen obligar a la inclusión de sofisticadas capacidades de control que, la mayoría de las veces, no son contempladas por ninguna herramienta de verificación.

- *No inclusión de la incertidumbre*: Como hemos visto anteriormente la inclusión de medidas de incertidumbre puede provocar que, situaciones que antes considerábamos anómalas, ahora sean perfectamente válidas. Sin embargo son pocos los trabajos que se encargan de verificar bases de conocimientos en presencia de incertidumbre. Entre ellos podemos destacar el trabajo de Wilkins y Buchanan (1986)

Para más información sobre las herramientas de verificación y su comparación se pueden consultar los trabajos de Murrell y Plant (1997), Gupta (1993) y López et al. (1990).

1.2. Validación de Sistemas Inteligentes

Una vez verificado el "software" del sistema, el proceso debe continuar con la validación del producto. Recordemos que validar un sistema inteligente supone analizar si los resultados del sistema son correctos (y por lo tanto se comporta como un experto más en un dominio de aplicación concreto), y si se cumplen las necesidades y los requisitos del usuario.

La validación puede verse desde dos ópticas diferentes: por un lado una *validación orientada a los resultados*, cuyo objetivo es comparar el rendimiento del sistema con un rendimiento esperado (proporcionado por una referencia estándar o por expertos humanos), y comprobar que el sistema alcanza un nivel que se considera aceptable. Por otro lado tenemos una la *validación orientada al uso*, que se centra en cuestiones que hacen referencia a la relación hombre-máquina, más allá de la corrección de los resultados obtenidos por el sistema.

Normalmente la validación orientada a los resultados constituye un prerrequisito para la realización de una validación orientada al uso. Así, si un sistema no presenta un rendimiento aceptable (o al menos indicaciones de que el rendimiento mejorará en un futuro al incluir mejoras en el desarrollo), los aspectos concernientes a la validación orientada al uso son irrelevantes. Por este motivo muchos autores incluyen la validación orientada al uso como una de las primeras fases de la evaluación, pasando el término "validación" a referirse únicamente a la validación orientada a los resultados. Esta es la aproximación que seguiremos a partir de ahora.

1.2.1 Principales características del proceso de validación

Al estudiar la validación nos damos cuenta de que no existe una clasificación global de los problemas a resolver ni tampoco existe una clara relación entre estos problemas y las técnicas destinadas a solucionarlos. Gupta (1993) señala que, entre los principales problemas existentes en la validación, caben destacar: la falta de métricas de evaluación prácticas y rigurosas; la falta de especificaciones, lo que conducen a evaluaciones subjetivas; y la falta de herramientas adecuadas.

El proceso de validación presenta distintos problemas para el ingeniero del conocimiento, que debe conocer las distintas aproximaciones que dispone para su eventual solución. Al respecto destacaremos los siguientes puntos:

- Personal involucrado en la validación.
- Partes del sistema a validar.
- Datos utilizados en la validación.
- Criterios de validación.
- Momento en el que se realiza la validación
- Métodos de validación
- Errores cometidos en la validación.

Personal involucrado en la validación

Una cuestión importante a determinar en todo tipo de validación es quién va a llevarla a cabo (Figura 1.2). El primer elemento a considerar es el ingeniero de conocimiento que ha desarrollado el sistema, ya que es quien mejor conoce las características del sistema inteligente. Sin embargo, incluir al ingeniero del conocimiento en el proceso puede afectar a la objetividad del mismo (ha dedicado mucho esfuerzo en el desarrollo del sistema y puede sentirse inclinado a sobrevalorar los resultados del mismo). De todas formas, en la validación siempre es necesaria la presencia de una persona que tenga un conocimiento amplio del sistema, aunque no sea su constructor.

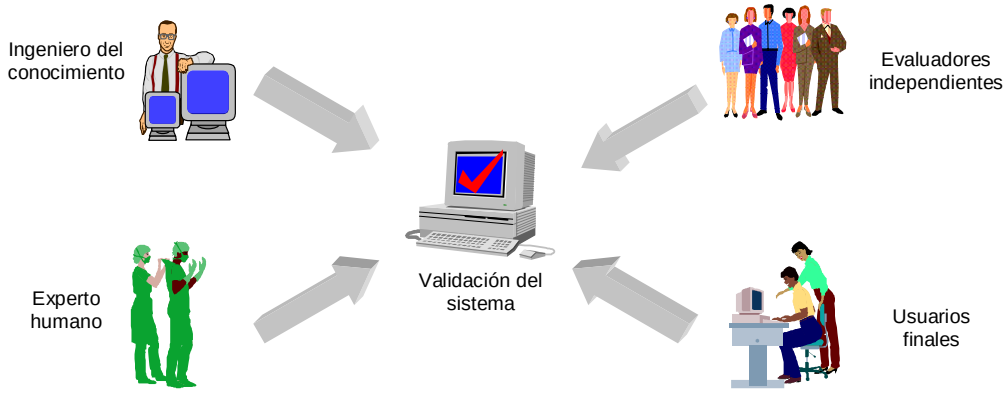


Figura 1.2. Personal involucrado en la validación de un sistema inteligente.

También es necesario contar con expertos humanos. Como veremos, el método básico para realizar la validación es el análisis de casos de prueba ya resueltos. Estos casos habrán sido analizados también por expertos humanos con los que podremos estudiar las discrepancias encontradas. Generalmente es conveniente que los expertos que participen en la validación no sean los mismos que colaboran en el desarrollo del sistema. Con esta medida se intenta conseguir que el conocimiento del sistema se adecue al de un consenso de expertos (y no sólo al conocimiento del experto colaborador en el diseño). No obstante, el tiempo de los expertos humanos es muy valioso, por lo que puede ser complicado contar con un amplio número de ellos para realizar una validación amplia y exhaustiva.

Debido a la necesidad de independencia en la validación surgió la idea de hacer recaer todas las responsabilidades en un grupo de expertos independiente (denominados "terceros expertos"). Sin embargo, si el constructor del sistema podía sobrevalorar el mismo, el uso de un grupo de validación totalmente independiente puede provocar el efecto contrario. Esta situación es la que Chandrasekaran (1983) describió como la "falacia del superhombre": se le exige más al sistema inteligente de lo que se le exigiría a un experto humano (teniendo en cuenta que el conocimiento del sistema inteligente es simplemente un modelo del conocimiento de los expertos humanos). También pueden aparecer problemas si los evaluadores no aceptan fácilmente la utilización de sistemas expertos en su área de trabajo, o si la solución propuesta pertenece a una "escuela de pensamiento" diferente a la suya. Como veremos posteriormente, para evitar subjetividades en el proceso de validación, se pueden llevar a cabo lo que se denominan "estudios ciegos".

Los usuarios finales del sistema también pueden participar en el proceso de validación; sin embargo, puede ocurrir que la experiencia de los mismos no sea suficiente para realizar la validación del sistema inteligente. Por ello generalmente su labor se destina a una validación orientada al uso.

Partes del sistema a validar

Nuestro principal objetivo es lograr que los resultados finales del sistema inteligente sean correctos. Sin embargo, también es interesante analizar si los resultados intermedios del sistema son correctos o si el razonamiento seguido hasta dar con la solución es apropiado.

La validación de los resultados intermedios puede ser interesante porque los resultados finales dependen de ellos. Así, el análisis de dichos resultados intermedios nos da una descripción del funcionamiento interno del sistema y nos permite una rápida corrección de los errores cometidos.

También es apropiado validar las estructuras de razonamiento; es decir, comprobar que el sistema alcanza la respuesta correcta por las razones correctas, ya que un proceso de razonamiento incorrecto puede provocar errores cuando queramos ampliar nuestra base de conocimientos. En este caso lo que se pretende es emular el proceso de razonamiento que realizan los expertos humanos. De esta forma los usuarios

del sistema encontrarán más agradable su utilización al seguir una línea lógica a la hora de plantear las cuestiones.

Para ver estas cuestiones con más claridad consideremos el siguiente ejemplo: sea un paciente en una unidad de cuidados intensivos, en la cual estamos monitorizando constantemente sus datos gasométricos. Además contamos con las características del contexto particular de su caso. Con estos datos intentamos hallar el estado de su balance ácido-base a través de un sistema inteligente. En la Figura 1.3 vemos que, ante un caso determinado, se ha producido un error y el resultado no es el esperado.

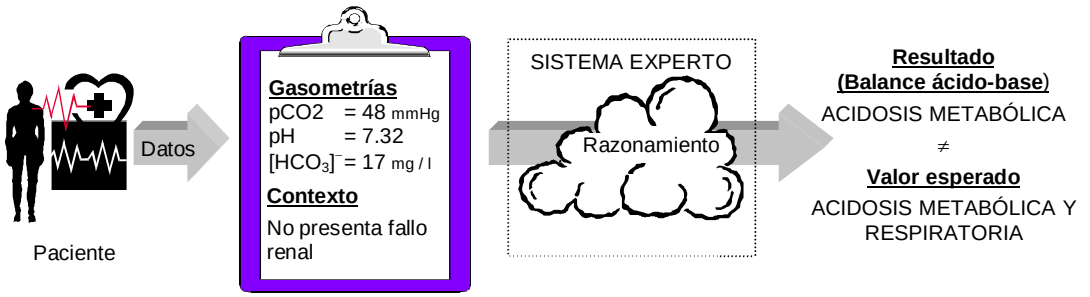


Figura 1.3. Las conclusiones finales del sistema sobre el "Balance ácido-base" no son correctas.

Analizando los resultados intermedios vemos que el error es debido a un fallo en la interpretación del pCO_2 debido a una error en una de las reglas que determina el estado del pCO_2 (Figura 1.4).

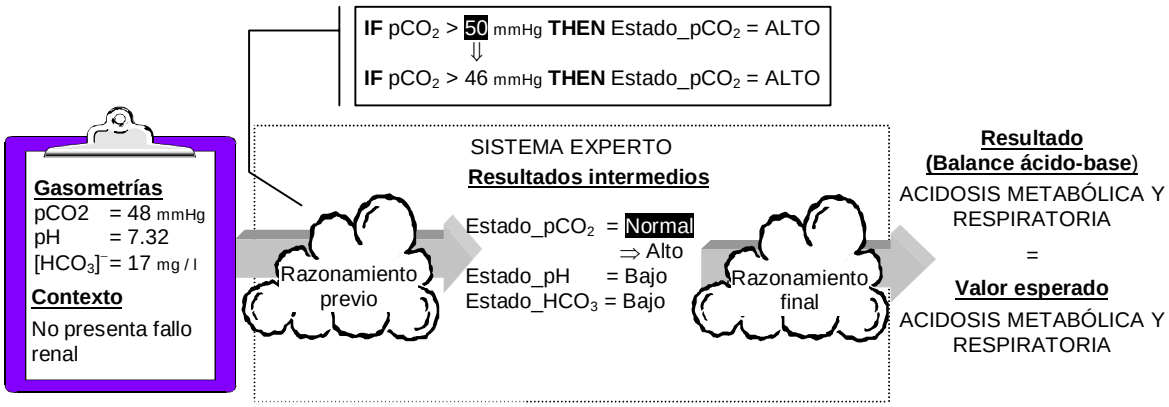


Figura 1.4. El error en la conclusiones finales era debido a que uno de los resultados intermedios (Estado_PCO_2) se interpretaba de forma errónea (se interpretaba como normal cuando debía interpretarse como alto).

Corregido el error las conclusiones del sistema son correctas. Sin embargo, si analizamos los procesos de razonamiento empleados vemos que en la determinación del estado del $[HCO_3]^-$ no se ha tenido en cuenta el hecho de que el paciente presente o no "Fallo Renal". La presencia de fallo renal puede alterar los valores "normales" del $[HCO_3]^-$, si en nuestro estudio no aparece ningún caso con esta enfermedad el sistema parecerá funcionar perfectamente, pero sus conclusiones serán erróneas en el momento que aparezca un paciente con fallo renal (Figura 1.5).

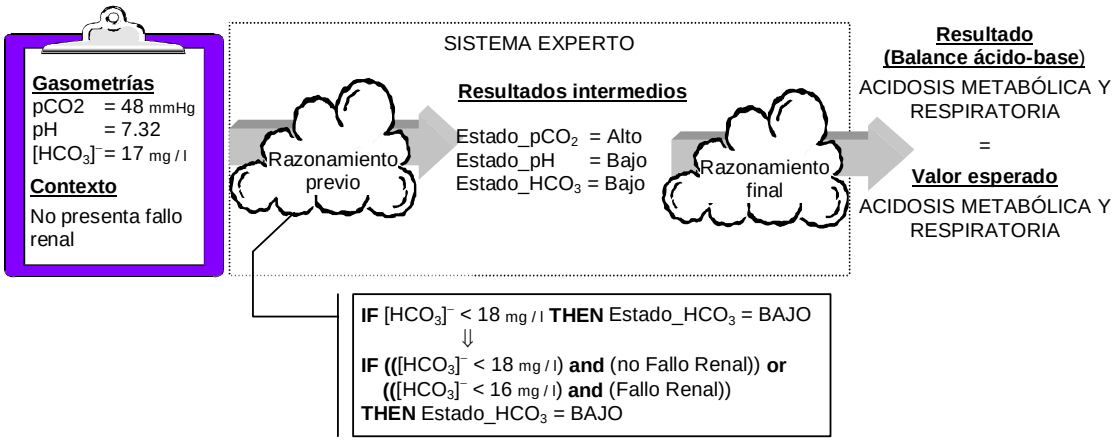


Figura 1.5. Los resultados son correctos pero en la determinación del estado del $[HCO_3]^-$ no se ha tenido en cuenta la presencia de fallo renal (para un paciente con fallo renal un $[HCO_3]^-$ de 17 mg /l sería normal y no bajo.)

Con este sencillo ejemplo hemos visto como el análisis de los resultados intermedios puede ayudar a la detección de errores en las conclusiones finales, y como una estructura de razonamiento inadecuada (en este caso incompleta) puede parecer correcta pero dar problemas cuando el ámbito de trabajo se amplía (aparecen pacientes con fallo renal).

Datos utilizados en la validación

El uso de casos de prueba es el método más ampliamente utilizado para la validación de sistemas expertos. En un mundo ideal contaríamos con una gran cantidad de casos que representarían un rango completo de problemas, y que son analizados por una serie de expertos. En la realidad, desafortunadamente, es muy común no disponer más que de un número reducido de casos y con poco expertos que nos ayuden a analizarlos. Para que una muestra de casos sea susceptible de ser aceptada en un proceso de validación debe cumplir dos propiedades fundamentales: cantidad y representatividad.

El número de casos empleados en la validación tiene que ser suficiente para que las medidas de rendimiento que obtengamos sean estadísticamente significativas. Ante esto podemos plantearnos un método muy sencillo de captura de datos: ir recogiendo todos los casos que podamos hasta que tengamos un número suficiente de ellos.

No obstante hay que considerar otra característica de la muestra como es su representatividad. No sólo hay que capturar un número elevado de casos, sino que éstos deber ser representativos de los problemas comunes a los que se va a enfrentar el sistema inteligente. Chandrasekaran (1983) aconseja que aquellos problemas que resuelva el sistema inteligente deben aparecer representados en los casos de prueba. O'Keefe et al. (1987) destacan que la cobertura de los casos es mucho más importante que su número, y que los casos deben representar con fiabilidad el dominio de entrada del sistema. El dominio de entrada está constituido por aquellos casos que son

susceptibles de ser tratados por el sistema inteligente, cuanto mayor sea el dominio de entrada más compleja se hace la validación del sistema.

Para intentar mantener la representatividad de los datos se suelen emplear muestreos estratificados. Así, por ejemplo, supongamos que tenemos un sistema inteligente médico a partir del cual pretendemos obtener tres posibles diagnósticos: A, B y C. Revisando la historia clínica comprobamos que en este tipo de clasificaciones el diagnóstico A ha aparecido el 80 % de las veces, B el 15 % y C el 5 %. De esta forma, si nuestra muestra está compuesta por 200 casos, 160 de ellos deben pertenecer al diagnóstico A, 30 al diagnóstico B y 10 al diagnóstico C.

Aunque el procedimiento de muestras estratificadas pueda parecer válido su utilización también ha sido objeto de controversias. Podemos poner el ejemplo de los sistemas expertos que analizan las probabilidades de bancarrota en firmas comerciales estadounidenses. En un año se producen sólo entre un 3 y un 5 % de bancarrotas, lo que significa que en una muestra estratificada una cantidad cercana al 96% lo constituirán casos pertenecientes a firmas comerciales que no han sufrido una bancarrota. Esta *inundación* de casos puede provocar que el sistema obtenga tasas de acierto elevadas aún cuando sus capacidades para predecir una bancarrota no sean adecuadas. En tales casos puede resultar adecuado establecer una muestra equilibrada, en la que el número de casos de bancarrota sea similar al número de casos en los que no se ha producido una bancarrota.

También puede ocurrir que el experto esté interesado en comprobar la respuesta del sistema ante casos extraños y complejos que, si empleáramos una muestra estratificada, aparecerían en una proporción minúscula. En tal caso la muestra debe variarse para acoger un número representativo de casos del tipo especificado.

Otro problema que puede aparecer es que no sea posible disponer de casos de prueba para validar el sistema. Hay que recordar que en la validación siempre es aconsejable no utilizar aquellos casos que se han utilizado en el diseño del sistema ya que, previsiblemente, el sistema habrá sido adaptado para tratar estos casos adecuadamente.

Una solución a la carencia de casos es la utilización de casos sintéticos, es decir, casos generados artificialmente por los expertos. El problema con esta aproximación es que demanda una considerable objetividad por parte de los validadores, para evitar generar casos que resalten los puntos fuertes del sistema.

A pesar de todos los problemas comentados, el estudio de casos de prueba resulta adecuado para la validación de sistemas expertos porque se adapta perfectamente a los métodos de desarrollo incremental. Podemos partir de un conjunto de casos limitado para realizar la validación de las primeras etapas de desarrollo y, a medida que el sistema se vaya ampliando y se desarrollen nuevas funcionalidades, podemos capturar nuevos casos de prueba para validar las nuevas capacidades del sistema. Después de una modificación importante podemos volver a analizar casos ya resueltos para comprobar si algún *efecto lateral* ha provocado errores que, antes de la modificación, no aparecían.

Criterios de validación

La casuística empleada en la validación del sistema debe incluir dos tipos de datos: por un lado las características de cada caso en particular y, por otro lado, un criterio que permita identificar el tipo de caso que estamos tratando. Siguiendo con el ejemplo del apartado 0, la casuística de validación incluiría como características de cada caso los valores del pH, del pCO_2, del $[HCO_3]^-$ y la presencia o no de fallo renal. Como criterio identificativo de cada caso en particular se incluye una etiqueta que asocia cada caso con la categoría a la que pertenece (como por ejemplo “acidosis metabólica y respiratoria).

El proceso de validación se haría de la siguiente manera (Figura 1.6):

(1) Se obtiene la casuística de validación.

(2) Los datos de la casuística son pasados al sistema inteligente que se encarga de interpretarlos.

(3) Los resultados del sistema y el criterio de validación que acompaña a los datos sirven de entrada para un proceso de validación en el que se analizará el rendimiento del sistema inteligente.

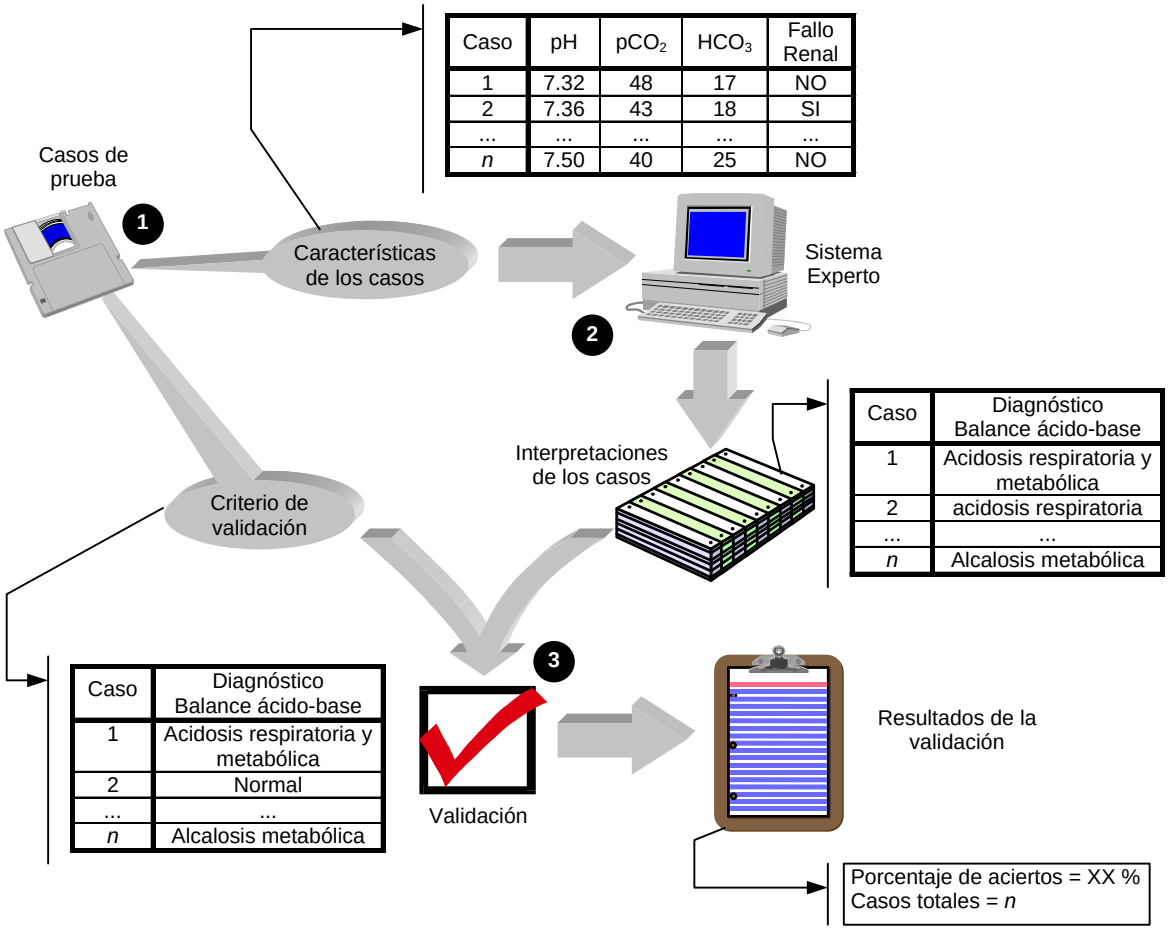


Figura 1.6. Proceso de validación a partir de casos de prueba: (1) Obtención de la casuística, (2) obtención de los resultados del sistema y (3) proceso de validación en el que se comparan los resultados del sistema con el criterio de validación..

Aunque el proceso pueda parecer sencillo existe un inconveniente importante: ¿qué utilizamos como criterio de validación?, ¿quién asocia cada caso de prueba con una categoría diagnóstica en particular?. Podemos diferenciar dos tipos de validación atendiendo al tipo de criterio establecido: validación contra el experto y validación contra el problema.

La **validación contra el experto** consiste, básicamente, en utilizar las opiniones y los diagnósticos de expertos humanos como criterio de validación. Este tipo de validación es la más comúnmente empleada en los sistemas inteligentes, después de todo, lo que pretendemos es construir un modelo del conocimiento del experto humano, por lo que resulta lógico utilizar a los expertos como criterio de nuestra validación.

Sin embargo, la validación contra el experto no está exenta de problemas, generalmente debidos a la propia naturaleza del conocimiento. Así, puede ser común que, expertos de un mismo nivel diagnostiquen soluciones diferentes ante el mismo problema. Incluso un mismo experto puede tener actitudes diferentes, ante un mismo

caso, según las condiciones del momento en que fue realizado el análisis. Las causas de estas discordancias pueden ser:

- Factores externos: estrés, cansancio, etc.
- Los expertos pueden no ser independientes y estar influidos por otro de mayor categoría profesional, o de mayor prestigio, o de mayor poder, etc.
- Ambigüedades o errores en la adquisición de los datos pueden provocar que los expertos den opiniones diferentes ante los mismos casos.
- Los expertos pueden pertenecer a diferentes escuelas de pensamiento.
- Tendencias, a favor o en contra, de los sistemas expertos pueden hacer variar las opiniones de los expertos humanos en la validación

Existen tres posibles tipos de validación contra los expertos: (1) validación contra un experto, (2) validación contra un grupo de expertos y (3) validación contra un consenso de expertos (Figura 1.7).

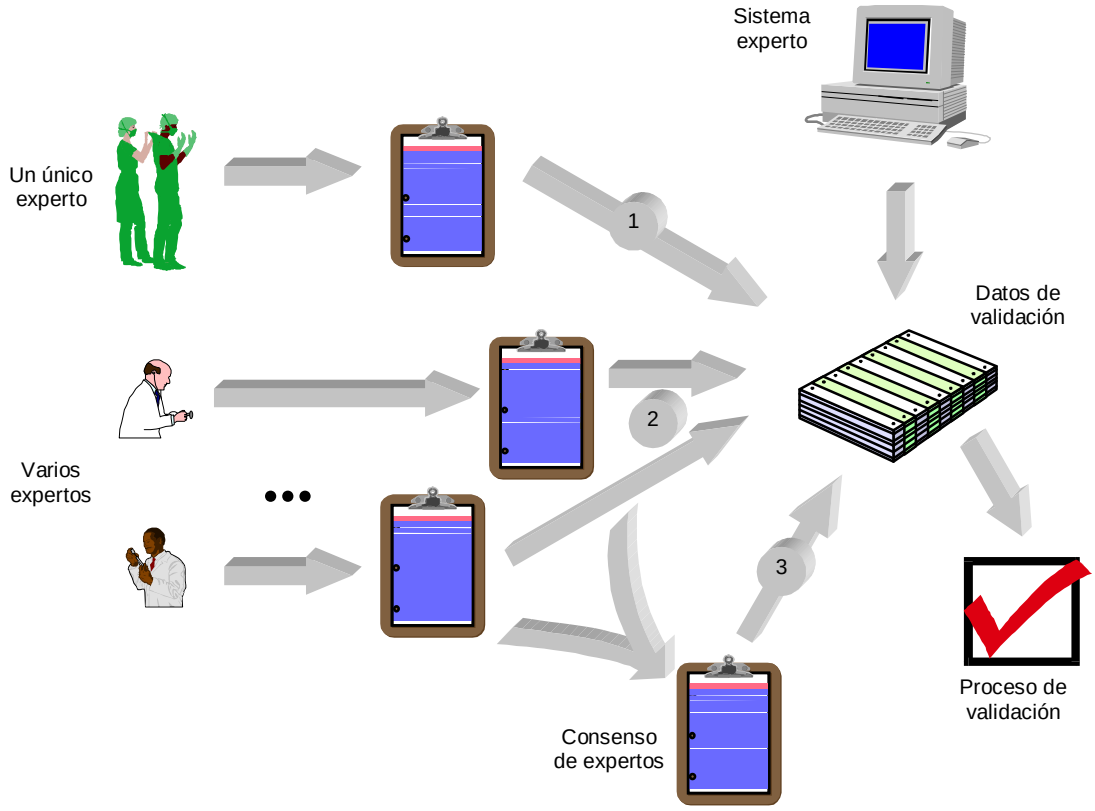


Figura 1.7. Posibles tipos de validación utilizando a los expertos como criterio: (1) con las opiniones de un único experto, (2) con las opiniones de varios expertos y (3) con las opiniones de varios expertos resumidas en un consenso..

La validación contra un único experto no es la más recomendable de todas pero, desgraciadamente, suele ser bastante común. Dada la escasa disponibilidad de expertos

humanos, no siempre es posible contar con varios expertos en el proceso de validación. El inconveniente de utilizar un único experto es que la objetividad del estudio es cuestionable.

Una situación más deseable a la hora de realizar la validación es contar con las opiniones de una serie de expertos humanos. Esto conlleva una serie de ventajas: (1) no estamos ligados a una única opinión, que puede ser errónea, y (2) permite comparar el grado de consistencia existente entre los expertos del dominio.

El principal inconveniente de esta técnica es cómo medir el rendimiento del sistema inteligente. Generalmente los expertos no suelen tener la misma cualificación y se suele buscar una concordancia elevada con aquellos expertos de mayor nivel. Sin embargo, si los expertos son todos de un nivel similar generalmente lo que se busca es comprobar que los diagnósticos del sistema se parezcan a los diagnósticos de los expertos, tanto como los diagnósticos de los expertos se parecen entre sí. Existen una serie de medidas y procedimientos estadísticos especialmente diseñados para medir estos acuerdos (medidas de Williams, análisis cluster, escalamiento multidiménsional y medidas de dispersion y tendencia) que analizaremos más adelante.

La otra opción comúnmente empleada en la validación con expertos, es conseguir unir las opiniones de varios expertos en una única opinión. Este consenso tiene la ventaja de que procura ser lo más objetivo posible, y si el acuerdo del sistema inteligente con el consenso es amplio, la confianza en el sistema aumentará considerablemente. El inconveniente de esta técnica es que, en cierta manera, estamos volviendo a la técnica de validación con un único experto, es decir, todo aquello que cae fuera del consenso es considerado erróneo. Sin embargo puede haber otras soluciones válidas que los expertos podrían haber elegido, pero que han cambiado para adaptarse a un estándar del cual no están plenamente convencidos (posiblemente influidos porque un experto de mayor nivel está de acuerdo con el consenso). Además, la búsqueda de un estándar o consenso entre los expertos puede ser una ardua tarea.

Entre los distintos métodos para lograr un consenso a partir de las opiniones de varios expertos destaca el método *Delphi* (Sackman, 1974). Este método se caracteriza por el anonimato y la interactuación remota de los participantes, su perfil retroalimentado y el uso de metodologías estadísticas en el análisis de los resultados, en el que se combinan generación de ideas y evaluación de opciones. Fue desarrollado por la compañía RAND como un método de prospección, y se basa en la recopilación de información cualitativa basada en juicios de expertos. El modelo aspira a eliminar los efectos indeseables de la interactuación directa eliminando el contacto personal entre los miembros del proyecto que, ni tan siquiera, conocen la identidad de los demás miembros del grupo.

Delphi utiliza un grupo o panel de expertos, seleccionados de acuerdo con su valía profesional y la naturaleza del problema, al que se envía un cuestionario completo para recopilar juicios acerca de procesos o fenómenos reales, más específicamente sobre su tendencia y desarrollo futuro. Todos los participantes formulan de manera secreta e independiente las hipótesis e ideas que les sugiere el problema, que son enviadas por escrito al coordinador del proyecto. Éste interpreta y consolida los resultados

preliminares - hipótesis, áreas de interés en relación al problema, propuestas - en un resumen global y anónimo de carácter estadístico y frecuentemente tabular.

Dicho resumen global se envía, junto con información estadística, a los expertos, a quienes se solicita que, en su caso, modifiquen sus apreciaciones iniciales o realicen las propuestas o aclaraciones que consideren oportunas, teniendo en cuenta las razones o consideraciones expuestas por los demás participantes. En la medida en que sus apreciaciones difieran de la opinión predominante del grupo se solicita que aclare su posición, lo que permite incorporar nuevas ideas y perspectivas al grupo de decisión. Tabulada la información, el coordinador envía un nuevo informe a los participantes y se reinicia el ciclo de consulta; a medida que éste se repite las posiciones de los expertos tienden a converger en las variables y procesos críticos del fenómeno, hasta alcanzar un grado suficientemente amplio de consenso. El proceso del método Delphi se representa en la Figura 1.8.

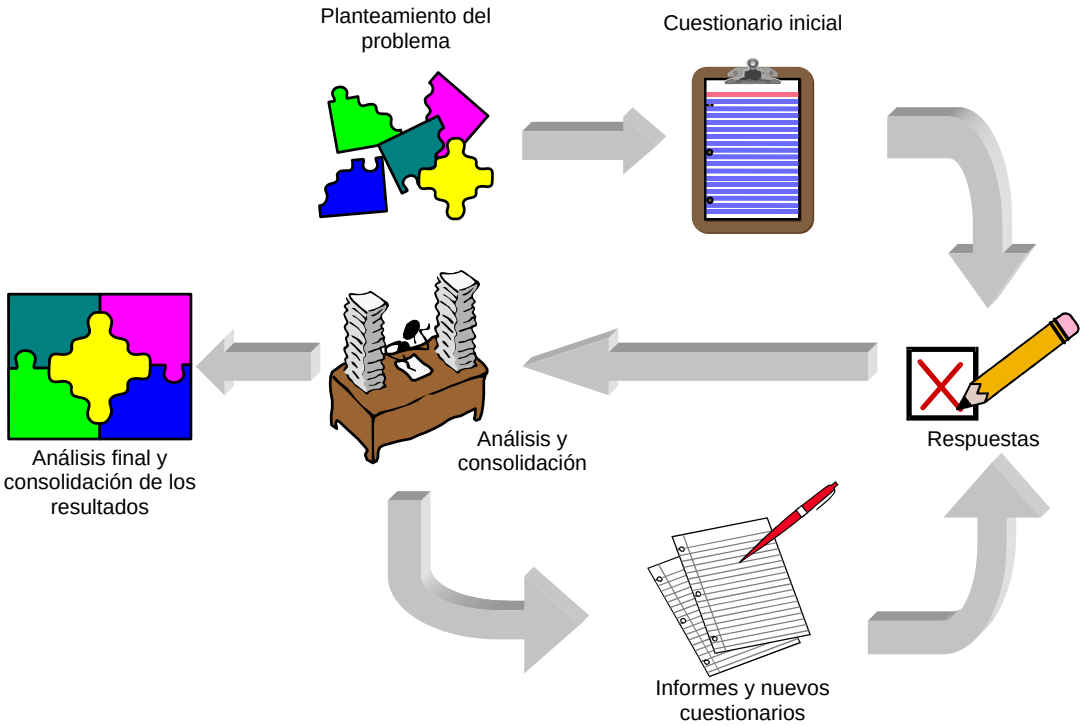


Figura 1.8. El proceso del método Delphi.

Este método se diferencia del panel de expertos tradicional por su carácter anónimo, y por la realización de dos o más ciclos iterativos de consulta, lo que le confiere gran potencia en cuanto a la generación de ideas y la orientación al compromiso lograda por la información estadística que se proporciona, al término de cada ciclo, a los expertos. Entre sus deficiencias más significativas cabe destacar el elevado grado de dependencia en relación al contenido, y la expresión de las preguntas en el cuestionario inicial y la selección de los expertos.

Existen otros muchos métodos para lograr un consenso entre varios expertos como el Brainstorming, las técnicas de grupos nominales, el AHP, ... (Medsker et al.,

1994). Tambien autores como (Xu et al., 1992) han propuesto métodos matemáticos para combinar información proveniente de distintas fuentes. Sin embargo el método más popular dentro de los sistemas expertos es el método Delphi debido a su carácter anónimo y a su método de realimentación controlada. Podemos encontrar ejemplos de la utilización del método Delphi en sistemas expertos en (Hamilton y Breslawski, 1996) y (Roth y Wood, 1993).

El otro tipos de validación atendiendo al tipo de criterio establecido es la **validación contra el problema**. Leyendo el apartado anterior, una pregunta que, probablemente haya surgido es: ¿qué pasa si los expertos humanos se equivocan?. Así, si puede verse como natural que dos expertos discrepen, también puede suceder que ninguno de ellos haya sabido dar con la solución real del problema.

Supongamos el ejemplo del sistema MYCIN (Shortliffe, 1976). MYCIN se encargaba de identificar la bacteria causante de la infección de un paciente y sugerir la terapia apropiada a cada caso. Evidentemente MYCIN puede evaluarse comparando sus diagnósticos con los de los expertos humanos. Pero también podemos comparar los diagnósticos de MYCIN con los resultados del laboratorio que nos identifican, de forma inequívoca, cual ha sido la bacteria causante de la infección.

Este segundo tipo de validación descrito se denomina *validación contra el problema*, ya que estamos tratando de descubrir si nuestro sistema resuelve realmente el problema que le han planteado.

La ventaja de este método de validación es clara: se trata de un método completamente objetivo, la solución real del problema es la que se muestra. Si nuestro sistema discrepa del experto pero coincide con la solución real, la credibilidad del sistema inteligente se verá aumentada.

Sin embargo este método también presenta inconvenientes. Uno de ellos es que podemos volver a caer en la “falacia del superhombre” que habíamos descrito con anterioridad, es decir, exigirle más al sistema inteligente de lo que se le exigiría al un experto humano. Así, supongamos un sistema que presenta un acuerdo del 70% con la solución real del problema. Este resultado puede parecer inadecuado, sin embargo, cuando analizamos los resultados de los distintos expertos humanos vemos que el acuerdo de estos con la solución real tampoco sobrepasa el 70% y que sus diagnósticos son muy similares a los del sistema inteligente. En tal caso podremos suponer que el comportamiento del sistema inteligente es aceptable.

Otro problema que surge en la validación contra el problema es que puede no ser posible obtener una solución real. Siguiendo con el ejemplo de MYCIN, el sistema inteligente aconsejaba una terapia para cada caso, la única forma de comprobar que la terapia es adecuada es probarla sobre el paciente. Evidentemente, por razones éticas, solo se podrá probar una terapia sobre un paciente si coincide con la terapia que ha prescrito el experto humano (lo que limita bastante el estudio). Además el hecho de que el paciente evolucione bien puede no ser indicativo de que la terapia aplicada es la mejor (puede existir otra que haga evolucionar al paciente más deprisa y con menos

sufrimiento). Por todos estos motivos la validación de MYCIN se realizó contra los expertos y no contra el problema (Yu et al., 1979).

O'Keefe et al. (1987) también recomiendan la validación contra expertos humanos, aunque indica que, si está disponible la solución real del problema, su utilización dentro del proceso de validación puede proporcionar información muy interesante.

Momento en el que se realiza la validación

Otro problema que surge a la hora de plantear la validación es: ¿cuándo realizarla?. Ante esto podemos encontrarnos dos posiciones: por un lado Bachant y McDermott (1984) advierten que validar un sistema que no está completo puede no ser útil, ya que éste no posee todo el conocimiento necesario para establecer decisiones correctas. Por otro lado Buchanan y Shortliffe (1984) recomiendan realizar la validación a lo largo de todo el desarrollo del sistema.

Como hemos visto al describir las distintas metodologías de la ingeniería del conocimiento, el punto de vista más comúnmente aceptado es el de realizar la validación a lo largo del desarrollo del sistema, realizando preferentemente un desarrollo incremental en el cual, al final de cada incremento, se realiza una validación. Sin embargo el razonamiento de Bachant y McDermott también es, en cierto sentido, válido. Así, en las primeras etapas de desarrollo, puede ser normal que el rendimiento del sistema no sea elevado y lo que se espera es que este rendimiento se eleve a medida que se va desarrollando el proyecto.

La validación que se realiza en etapas tempranas del desarrollo esta muy vinculada al proceso de adquisición del conocimiento. Así surge un nuevo proceso, denominado *refinamiento del conocimiento,* y que podemos encuadrar dentro de la fase de adquisición. El proceso de refinamiento consiste en verificar y validar el conocimiento recién adquirido en busca de problemas, resultados incorrectos, estructuras inadecuadas, etc. Existen herramientas como SEEK (Politakis, 1985) y SEEK2 (Ginsber y Weiss, 1985) que se encargan de verificar y validar el nuevo conocimiento adquirido identificando las reglas que pueden ser causa de los errores.

Otro aspecto a tener en cuenta, y que guarda cierta relación con el momento de realizar la validación, es la diferenciación existente entre la llamada validación retrospectiva y la validación prospectiva. La *validación retrospectiva* se realiza sobre casos históricos ya resueltos y almacenados en una base de datos. Este tipo de validación es la más comúnmente realizada en los sistemas expertos y los casos utilizados pueden incluir como referencia de validación tanto opiniones de expertos humanos, como la solución real al problema planteado (validación contra expertos o contra el problema). La validación retrospectiva se utiliza en las etapas de desarrollo del sistema, antes de que este se instale en su campo de trabajo habitual.

Por otro lado, la *validación prospectiva* consiste en confrontar al sistema con casos reales y ver si es capaz de resolverlos o no (está frecuentemente relacionada con la validación orientada al problema). En la validación prospectiva no se utilizan casos

almacenados en bases de datos sino que se utilizan casos que en ese momento están siendo tratados por expertos humanos. De este modo se puede evaluar, no sólo la corrección de los resultados, sino aspectos referentes al uso del sistema. El problema surge, al igual que en la validación contra el problema, cuando el dominio de aplicación es crítico y el sistema intenta manipular el entorno (por ejemplo, administrándole una terapia a un paciente). Si la manipulación no ha sido aprobada por un experto humano no podrá llevarse a cabo, lo que puede limitar bastante este tipo de validación.

La validación prospectiva se utiliza una vez que hemos validado el sistema en un entorno de desarrollo y utilizando casos históricos, y se desea realizar una nueva validación en el campo de aplicación del sistema. Este tipo de validación es similar a las pruebas beta que analizaremos más adelante.

Errores en la validación

En el proceso de validación se pueden dos tipos de errores: de Tipo I o de riesgo para el desarrollador y de Tipo II o de riesgo para el usuario (Tabla 1.3).

		Estado del sistema inteligente	
		El sistema es válido	**El sistema NO es válido**
Acción	**El sistema se acepta como válido**	Decisión correcta	Error Tipo II (riesgo para el usuario)
	El sistema NO se acepta como válido	Error Tipo I (Riesgo para el desarrollador)	Decisión correcta

Tabla 1.3. Posibles errores en el proceso de validación.

Los errores de Tipo I se producen cuando un sistema es considerado como no válido, aun a pesar de ser válido. Este error aumenta innecesariamente los costes de desarrollo del sistema y merma la credibilidad en el mismo. Se denominan como de "riesgo para el desarrollador" porque el propio desarrollo del sistema puede ponerse en entredicho.

Por otro lado, los errores de Tipo II se producen cuando se acepta como válido un sistema que no lo es. Las consecuencias de este error son más peligrosas que las del error de Tipo I, sobre todo si el sistema actúa en dominios críticos (un sistema inteligente médico que diagnostique una enfermedad incorrectamente puede provocar a los pacientes un sufrimiento innecesario).

Métodos de validación

Los métodos para realizar la validación pueden dividirse en dos grupos principales: métodos cualitativos y métodos cuantitativos. Los métodos cualitativos emplean técnicas subjetivas de comparación del rendimiento mientras que los métodos cuantitativos se basan en la utilización de medidas estadísticas. Esto no implica que los métodos cualitativos sean menos formales que los métodos cuantitativos. Estas técnicas

no son mutuamente excluyentes y lo normal es utilizar una combinación de las mismas. Dentro de los **métodos cualitativos** de validación podemos destacar los siguientes:

La **validación de superficie** es un proceso informal en el que expertos colaboradores, e ingenieros de conocimiento, discuten y analizan la validez de las conclusiones obtenidas por el sistema inteligente. Esta técnica es útil para evaluar módulos concretos durante la fase de desarrollo, pero no es conveniente para analizar el comportamiento global del sistema, ya que el procedimiento no es susceptible de ser formalizado.

La **prueba de Turing** puede definirse como una técnica de validación múltiple, en la que un experto de referencia -el evaluador- debe analizar un conjunto de información y datos, que previamente han sido interpretados -de forma ciega e independiente- por el sistema inteligente y por un grupo de expertos. La información, los datos, y las interpretaciones efectuadas por todos los miembros involucrados en el estudio, son presentadas al evaluador sobre un mismo soporte físico y con el mismo formato. De este modo, si el sistema estuviese correctamente diseñado y construido, y los expertos fuesen totalmente independientes, el evaluador no debería poder identificar ni al sistema ni a ninguno de los expertos.

Los **tests de campo** consisten en colocar al sistema inteligente en el que va a ser su entorno de trabajo habitual y permitir que los usuarios interaccionen con él en busca de posibles errores. Es lo que en la ingeniería del software conocíamos como pruebas beta.

Los tests de campo presentan una serie de ventajas como: (1) parte de las tareas de validación se efectúan por los usuarios del sistema, (2) el nivel de rendimiento aceptable se obtiene implícitamente (cuando los usuarios dejan de notificar problemas) y, (3) permite descubrir errores que se habían pasado por alto en otro tipo de validaciones.

Sin embargo su utilización conlleva una serie de problemas: (1) los usuarios pueden inundarnos con llamadas sobre preguntas menores que tienen poca relación con el rendimiento del sistema en si, (2) el sistema puede perder credibilidad si el prototipo mostrado es muy incompleto, y (3) sólo puede utilizarse en aquellos dominios no críticos en los que los usuarios están capacitados para comprobar la corrección de las conclusiones del sistema inteligente.

Un ejemplo de la utilización de los tests de campo puede verse en la validación del sistema R1/Xcon (Bachant y McDermott, 1984)

La **validación de subsistemas** requiere la división de la base de conocimientos en diversos subsistemas o módulos que, posteriormente, se validan por separado utilizando otros métodos.

Esta técnica de “divide y vencerás” permite reconocer más fácilmente los errores y facilita el proceso de validación. Sin embargo, presenta una serie de inconvenientes como son: (1) no todos los sistemas se pueden dividir fácilmente en subsistemas independientes y, (2) la validación de todos los subsistemas por separado no es

equivalente a la validación del sistema completo. Por ejemplo, supongamos dos módulos de un sistema inteligente médico que diagnóstican por separado la administración de dos drogas distintas. La administración de las drogas por separado no ofrece problemas, sin embargo, su administración conjunta puede ser peligrosa para la vida del paciente.

El **análisis de sensibilidad** consiste en presentar, a la entrada del sistema, una serie de casos muy similares entre sí, conteniendo sólo pequeñas diferencias. El impacto de dichas variaciones en los casos de entrada puede ser estudiado observando los cambios resultantes en la salida.

Esta técnica es especialmente útil cuando tratamos sistemas que manejan medidas de incertidumbre, ya que puede estudiarse el impacto de los cambios en dichas medidas, tanto en los resultados intermedios, como en las conclusiones finales.

Los **grupos de control** se basan en el hecho de que los sistemas expertos pretenden simplificar el trabajo a realizar por parte de los expertos humanos. Por ello, no sólo debe evaluarse el sistema por separado, también es útil comprobar el impacto que tiene el sistema en la organización.

En este caso se presentan los casos a dos grupos de expertos, unos que utilizan el sistema inteligente y otros que trabajan sin él (y constituyen el denominado grupo de control). De esta forma podemos comparar el rendimiento de los expertos cuando utilizan el sistema inteligente, y cuando no lo utilizan. Para una mayor discusión sobre los grupos de control y otras técnicas denominadas *quasi-experimentales* se puede consultar (Adelman, 1991).

1.3. Métodos Cuantitativos de Validación

Debido a su extensión dedicaremos un apartado completo a describir los principales métodos cuantitativos de validación. A diferencia de los métodos cualitativos, los métodos cuantitativos de validación están basados en análisis estadísticos, que tratan de comparar las conclusiones del sistema inteligente con las producidas por los expertos del dominio. Existen muchas técnicas estadísticas susceptibles de ser usadas en un proceso de validación, en este trabajo comentaremos las más usadas en la bibliografía que se subdividen en tres grupos: medidas de pares, medidas de grupo y ratios de acuerdo.

1.3.1 Medidas de pares

Uno de los métodos de validación contra el experto consiste en comparar las interpretaciones del sistema con las interpretaciones de un único experto humano. Estas comparaciones se realizan a través de medidas que involucran a pares de expertos y cuyo proceso de realización es el siguiente (Figura 1.9): (1) a partir de los datos incluidos en la base de datos de validación se desarrolla una tabla o matriz de contingencia para cada uno de los posibles pares que se puedan formar entre los

expertos involucrados en la validación (dentro de los cuales incluimos al sistema inteligente), y (2) se extrae de la tabla de contingencia la medida de pares determinada.

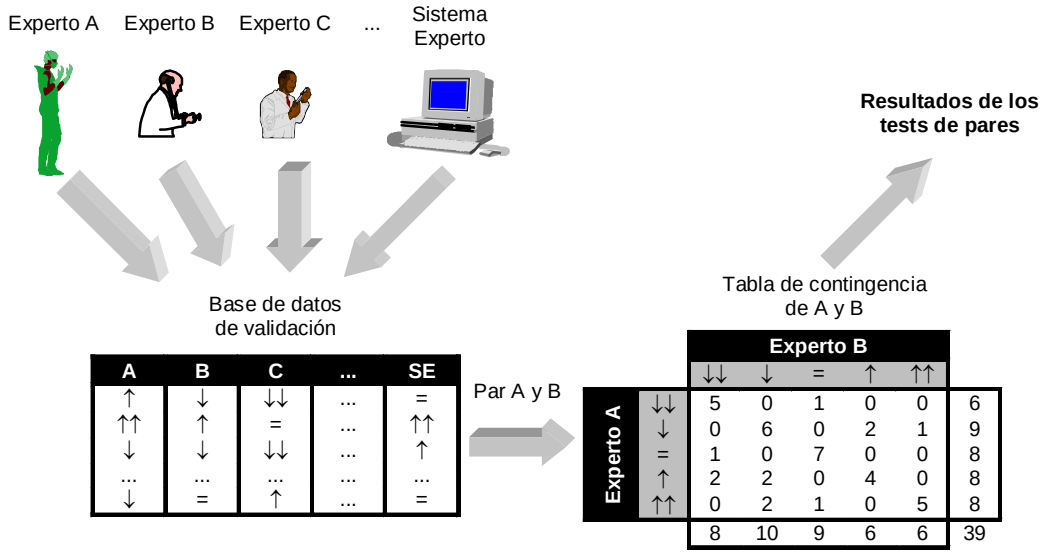


Figura 1.9 Proceso de realización de los tests de pares.

Las medidas de pares se dividen en dos grupos, medidas de acuerdo y medidas de asociación.

Las **medidas de acuerdo** nos dan un índice que cuantifica las coincidencias entre las interpretaciones de dos expertos. Entre las medidas de este tipo más importantes podemos destacar: (a) el índice de acuerdo, (b) el índice de acuerdo dentro de uno, (c) kappa, y (d) kappa ponderada.

El **índice de acuerdo**, es el cociente entre el número de observaciones de acuerdo y el número de observaciones totales, como se muestra en la ecuación (1.1) en la que N es el número total de casos, k el número total de categorías semánticas y n_{ij} el número total de casos en la celda ij de la tabla de contingencia.

$$\text{Índice de acuerdo} = \frac{\sum_{i=j}^{k} n_{ij}}{N} = \sum_{i=j}^{k} p_{ij} \tag{1.1}$$

La principal ventaja de esta medida es la sencillez de su interpretación, que ha hecho que su uso se extienda en distintos campos y aplicaciones. Sin embargo tiene el inconveniente de no diferenciar los desacuerdos según su importancia y el de no tener en cuenta la casualidad.

El **índice de acuerdo dentro de uno**, es similar al índice anterior pero considera acuerdos parciales aquellos acuerdos que se diferencian en una única categoría semántica como vemos en la ecuación (1.2).

$$\text{Índice de acuerdo dentro de uno} = \frac{\sum_{\substack{i=j \\ i=j\pm1}}^{k} n_{ij}}{N} = \sum_{\substack{i=j \\ i=j\pm1}}^{k} p_{ij} \tag{1.2}$$

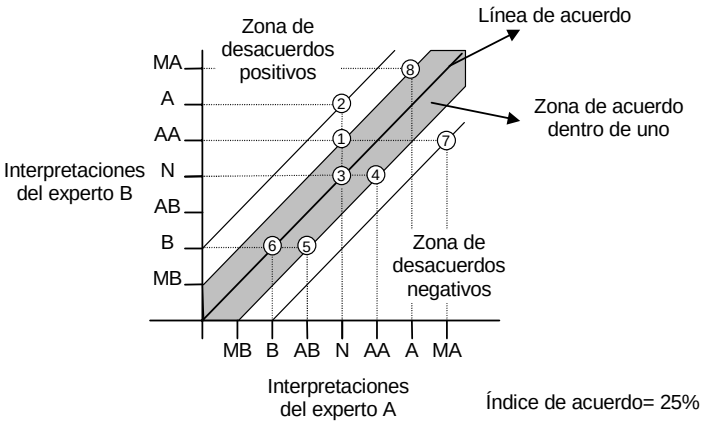
La ventaja de utilizar esta aproximación es que, en cierta forma, elimina los problemas asociados a las categorías semánticas ordinales cuyos límites no pueden establecerse con claridad (Figura 1.10). Además puede permitir el análisis de tendencias en las interpretaciones si distinguimos entre los acuerdos dentro de uno "optimistas" – que se producen por encima de la diagonal de acuerdo – o "pesimistas" – que se producen por debajo de la diagonal de acuerdo.

Escala semántica para la clasificación simbólica de una variable	Muy Bajo (MB)	Bajo (B)	Algo Bajo (AB)	Normal (N)	Algo Alto (AA)	Alto (A)	Muy Alto (MA)

(a)

Cases	***Expert A***	***Expert B***
1	Normal	Algo Alto
2	Normal	Alto
3	Normal	Normal
4	Algo Alto	Normal
5	Algo Bajo	Bajo
6	Bajo	Bajo
7	Muy Alto	Algo Alto
8	Alto	Muy Alto

(b)



(c)

Figura 1.10 (a) Escala semántica para la clasificación simbólica de una determinada interpretación, (b) Datos de ejemplo, (c) Representación de las zonas de acuerdo y desacuerdo y valores de los índices de acuerdo.

Como inconvenientes de esta medida podemos citar en primer lugar que, en caso de que los acuerdos sean elevados, o que las categorías semánticas sean pocas, el índice presenta una tendencia a adoptar valores cercanos a la unidad. Por otro lado, al igual que el índice de acuerdo simple, no presenta ningún ajuste para corregir aquellos acuerdos debidos a la casualidad.

Para corregir los acuerdos debidos a la casualidad Cohen (1960) propuso la medida **kappa**. Esta medida es aplicable tanto a escalas ordinales como a escalas nominales y está basada en dos cantidades: p_o (proporción de acuerdo observado) y p_c

(proporción de acuerdo esperado debido a la casualidad). De esta forma, $1 - p_c$ representa el máximo acuerdo posible una vez que se ha eliminado la casualidad y $p_o - p_c$ representa el acuerdo obtenido una vez que se ha eliminado la casualidad. Esto nos permite definir el índice kappa según la ecuación (1.3):

$$\kappa = \frac{p_o - p_c}{1 - p_c} \tag{1.3}$$

El término p_o es la proporción de acuerdo vista anteriormente, mientras que el término p_c es la suma de los productos de las proporciones marginales correspondientes a la diagonal principal y se expresa mediante la ecuación (1.4):

$$p_c = \sum_{i=j}^{k} p_{i.} p_{.j} \tag{1.4}$$

En esta ecuación se ha utilizado la notación de puntos, en la que un punto en un subíndice indica que se realiza una suma sobre dicho subíndice.

El problema de kappa es que trata todos los desacuerdos de la misma forma, dándoles la misma importancia. Por este motivo Cohen (1968) desarrollo una modificación de la kappa original que denomina kappa ponderada.

Por último, **kappa ponderada** (κ_w) es una medida de acuerdo que corrige aquellos acuerdos debidos a la casualidad, y pondera de forma distinta los desacuerdos encontrados. La ponderación se hace a partir de una matriz de pesos en la que, para cada posible par de categorías *ij*, se define un peso v_{ij}, que cuantifica el desacuerdo existente. A las celdas pertenecientes a la diagonal principal (que representan el acuerdo perfecto) se les suele asignar el valor 0 indicando que no existe ningún desacuerdo. El mayor valor de desacuerdo v_{max} es fijado por el investigador. Para cualquier conjunto de pesos, kappa ponderada es invariable ante transformaciones multiplicativas positivas, es decir, que kappa ponderada no cambiará de valor si sus pesos se multiplican por un valor mayor que cero.

La definición de kappa ponderada se muestra en la ecuación (1.5), en la que p_{oij} es la proporción de acuerdo observada para la casilla *ij*, p_{cij} es la proporción de acuerdo debido a la casualidad correspondiente a la casilla *ij*, v_{ij} es el peso correspondiente a la casilla *ij*, y *k* es el número de categorías.

$$\kappa_w = 1 - \frac{\sum_{i=1,\, j=1}^{k} v_{ij}\, p_{oij}}{\sum_{i=1,\, j=1}^{k} v_{ij}\, p_{cij}} \tag{1.5}$$

Existen ocasiones en las que valores elevados en los índices de acuerdo se corresponden con valores bajos en las medidas kappa. Como señalan Donker et al. (1992) esto suele ser debido a que la casuística está poco balanceada y la mayoría de los casos se concentran en unas pocas categorías, provocando que el acuerdo debido a la casualidad sea elevado.

Kappa ponderada también es aplicable a escalas nominales y ordinales. En las escalas ordinales los pesos de las discrepancias suelen ser fáciles de asignar ya que se basan en el propio orden de las categorías semánticas. Sin embargo, en escalas nominales, la asignación de pesos requiere un estudio más detallado.

Las **medidas de asociación** nos miden el grado de asociación lineal existente entre el sistema y el experto humano. No debe confundirse asociación con acuerdo ya que las interpretaciones de dos expertos pueden estar asociadas pero no en la dirección del acuerdo. Estas medidas sólo son aplicables en el caso de que las categorías semánticas de la interpretación sigan una escala ordinal, ya que presuponen la existencia de un orden.

Dentro de las medidas de asociación nos encontramos (a) la tau de Kendall y sus variantes (tau b de Kendall y Gamma de Goodman-Kruskal), y (b) la rho de Spearman. Estas medidas reproducen las características de la medida de asociación más popular: el coeficiente de correlación lineal (también conocido como el coeficiente de correlación producto-momento de Pearson), pero añaden una nueva característica: la invariabilidad ante aquellas transformaciones para las cuales se mantiene el orden de magnitud (algo deseable cuando se trabaja con escalas ordinales).

La **tau de Kendall** (Kendall & Gibbons, 1990) se define según la ecuación (1.6), en la que C representa el número de observaciones concordantes, D el número de observaciones discordantes y n el número total de casos.

$$\tau = \frac{C - D}{n(n-1)/2} \tag{1.6}$$

Decimos que un par de observaciones (x_i, y_i) y (x_j, y_j), son concordantes cuando cumplen que $(x_i - x_j)\,(y_i - y_j) > 0$. De la misma forma dos pares de observaciones son discordantes cuando cumplen que $(x_i - x_j)\,(y_i - y_j) < 0$.

El problema de tau se presenta con aquellos pares que se consideran ligados, en los cuales $(x_i - x_j)\,(y_i - y_j) = 0$. En la ecuación (1.6) estos pares se consideran en el denominador, pero no en el numerador. En entornos de validación es común que el número de pares ligados sea muy elevado, por lo que el valor de la tau de Kendall queda muy distorsionado. Para evitar esto se ha propuesto modificar el denominador para tener en cuenta las ligaduras – definiendo la tau b de Kendall que vemos en la ecuación (1.7) en donde U representan los pares ligados en x y V los pares ligados en y – o eliminado directamente las ligaduras del denominador – definiendo la gamma de Goodman-Kruskal que vemos en la ecuación (1.8).

$$\tau_b = \frac{C - D}{\sqrt{\left[\frac{n(n-1)}{2} - U\right]\left[\frac{n(n-1)}{2} - V\right]}} \tag{1.7}$$

$$\gamma = \frac{C - D}{C + D} \tag{1.8}$$

Aunque la interpretación de tau es muy sencilla, el tratamiento que hace de las observaciones ligadas (incluyendo las variantes de tau b y gamma) resulta muy artificioso en entornos de validación, por lo que se suele sugerir el empleo de la rho de Spearman.

La **rho de Spearman** (Kendall & Gibbons, 1990) se representa habitualmente como r_s y básicamente es un coeficiente de correlación basado en rangos, y no en valores. En la ecuación (1.9) vemos la definición de la rho de Spearman, que coincide con el coeficiente de correlación lineal salvo en el hecho de que los pares de valores (x, y) se han convertido en pares de rangos (R, S). Un rango no es más que un número de orden que se le asigna a cada valor, en caso de existir ligaduras se asigna a cada miembro del grupo ligado el promedio de los rangos que se habrían asignado de no haber estado ligados. De esta forma se consigue una medida de asociación, similar al coeficiente de correlación lineal pero invariable ante transformaciones que mantienen la relación de orden entre los datos.

$$r_s = \frac{\sum_{i=1}^{n}\left(R_i - \overline{R}\right)\left(S_i - \overline{S}\right)}{\sqrt{\sum_{i=1}^{n}\left(R_i - \overline{R}\right)^2 \sum_{i=1}^{n}\left(S_i - \overline{S}\right)^2}} \tag{1.9}$$

La interpretación de la rho de Spearman no es tan sencilla como la de tau pero su tratamiento de las ligaduras parece más comprensible dentro del contexto de validación. Además esta medida también tiene en cuenta la distancia existente entre las distintas categorías semánticas (algo que tau no contempla).

1.3.2 Medidas de grupo

Las medidas de pares son útiles cuando el número de expertos es reducido. Sin embargo si la validación involucra un grupo amplio de expertos la información que proporcionan las medidas de pares puede resultar difícil de interpretar.

En todo caso, las medidas de pares pueden servir de base para otro tipo de medidas: las medidas de grupo, cuyo objetivo es analizar conjuntamente las interpretaciones de los expertos y tratar de buscar estructuras de representación que permitan una interpretación más sencilla dentro del contexto de la validación.

El procedimiento para obtener las medidas de grupo es el siguiente: (Figura 1.11) (1) se obtienen las medidas de pares para cada uno de los posibles pares de expertos de nuestra validación, (2) se agrupan los resultados de cada medida de pares en una tabla resumen y, (3) se obtiene la medida de grupo a partir de los datos incluidos en las tablas resumen.

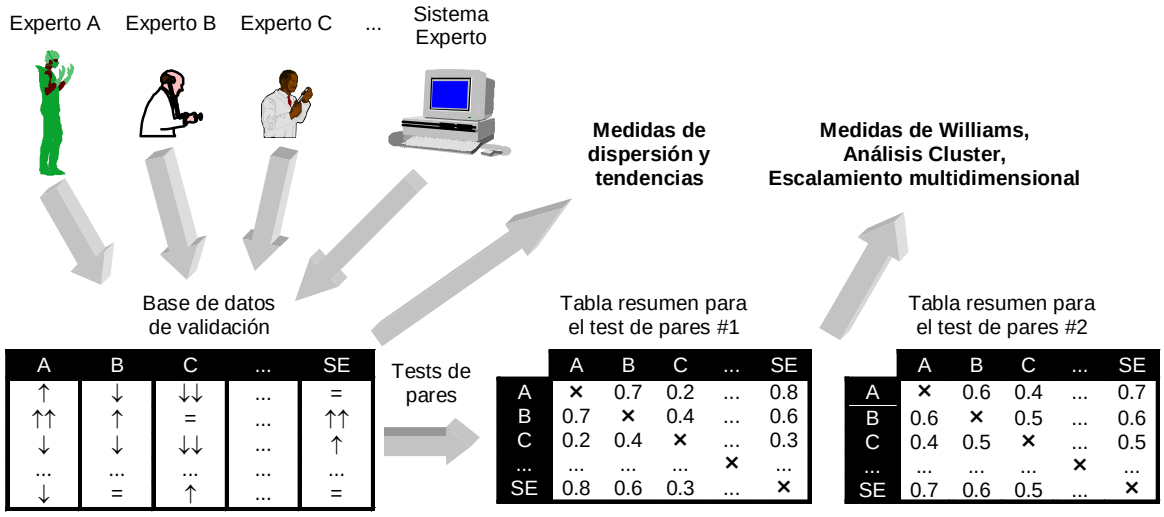


Figura 1.11. Proceso de realización de los tests de grupo.

Dentro de las medidas de grupo podemos citar: (a) el índice de Williams, (b) el análisis cluster, (c) el escalamiento multidimensional y (d) las medidas de dispersión y tendencia.

El **índice de Williams** (1976) se representa en la ecuación (1.10). En esta expresión P_0 representa el acuerdo existente entre un experto aislado en relación a un grupo de expertos de referencia, mientras que P_n representa el acuerdo existente dentro de dicho grupo de referencia. P_0 se define en la ecuación (1.11) y P_n en la ecuación (1.12) en las que n representa el número de expertos de referencia y $P_{(a,\,b)}$ una medida de pares que relaciona las interpretaciones de los expertos a y b.

$$I_0 = \frac{P_0}{P_n} \tag{1.10}$$

$$P_0 = \frac{\sum_{a=1}^{n} P_{(0,a)}}{n} \tag{1.11}$$

$$P_n = 2\frac{\sum_{a=1}^{n-1}\sum_{b=a+1}^{n} P_{(a,b)}}{n(n-1)} \tag{1.12}$$

La interpretación de I_n es la siguiente:

- Si I_n es menor que uno indica que el acuerdo entre el experto aislado y el grupo de expertos es menor que el acuerdo entre los propios miembros del grupo.

- Si I_n es igual a uno indica que el experto aislado coincide con el grupo al mismo nivel que los miembros del grupo coinciden entre sí.

- Si I_n es mayor que uno podríamos decir que el experto aislado coincide con el consenso del grupo de expertos.

El inconveniente de estas medidas es que pueden ser mal interpretadas si dentro del grupo de referencia existe un experto claramente en desacuerdo con los demás, o si el acuerdo dentro del grupo de referencia es escaso. En tales casos se debería estudiar si el experto en desacuerdo debería ser apartado del grupo de referencia o tratar de utilizar técnicas para el desarrollo de un consenso dentro del grupo.

El **análisis cluster** (Everitt, 1993) tiene como objetivo establecer grupos de expertos, según su grado de concordancia, e identificar a qué grupo se parece más nuestro sistema inteligente. Los métodos que podemos emplear para realizar un análisis clúster pueden ser de dos tipos: jerárquicos y no jerárquicos.

La aplicación de un método jerárquico de análisis clúster implica la construcción de una matriz de concordancia que describa *distancias* entre todos los miembros involucrados en el estudio. Una distancia apropiada podría ser, por ejemplo, los índices de acuerdo encontrados entre los distintos expertos. A partir de los datos de la matriz de concordancia podemos establecer una secuencia de *agrupamientos anidados*, que definen una estructura en árbol denominada *dendrograma*, en la que cada nivel representa una partición del conjunto global de los elementos que son objeto del análisis. El algoritmo de análisis cluster jerárquico se conoce habitualmente con las siglas SAHN (Sequential, Agglomerative, Hierarchical and Nonoverlaping) y lo podemos encontrar en (Dubes, 1993).

En el ejemplo de la Figura 1.12 se muestra un dendrograma típico del análisis cluster jerárquico. Si cortamos el dendrograma por la línea de trazos obtenemos los siguientes tres clusters: (ES, B, A), (D) y (C).

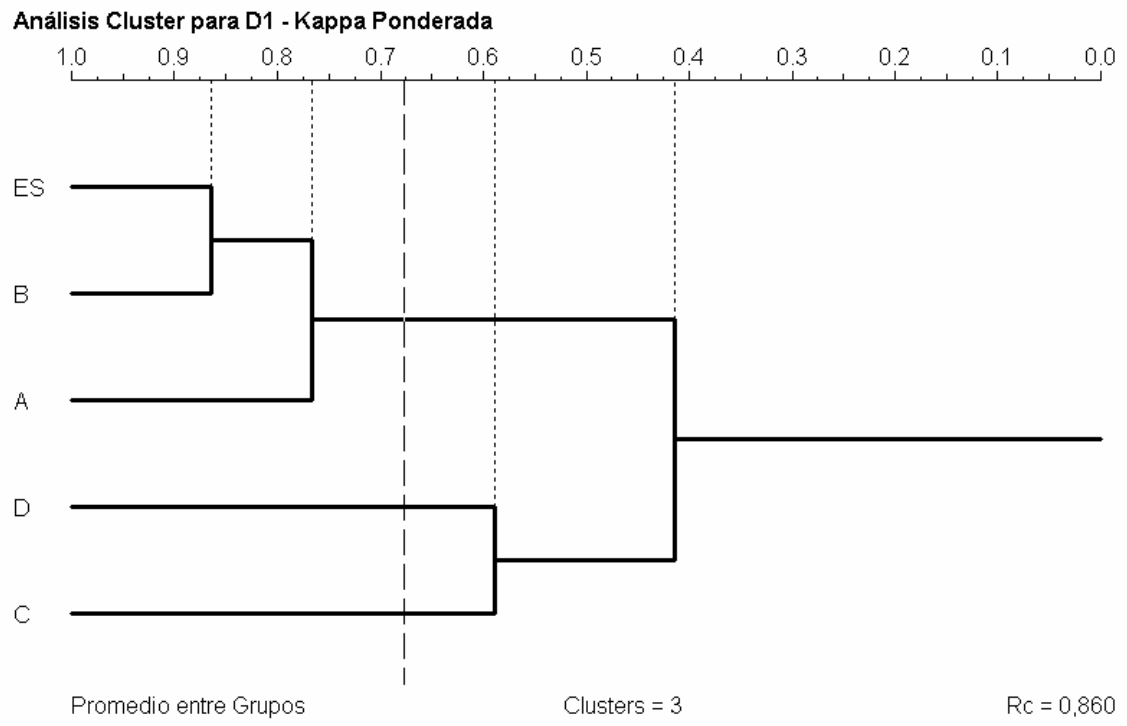


Figura 1.12 Dendrograma del análisis cluster

Para validar los resultados del análisis cluster se utiliza el coeficiente de correlación cofenética que no es más que el grado de correlación existente entre las distancias iniciales existentes entre los expertos y las distancias que se extraen del dendrograma.

El análisis cluster jerárquico presenta las siguientes ventajas: (1) es rápido, siempre y cuando el número de expertos no sea muy elevado, (2) permite obtener una visión de conjunto de las similitudes entre los distintos expertos y, (3) permite obtener distintas particiones del conjunto de expertos (simplemente cortando el dendrograma por distintos puntos).

Sin embargo, las técnicas jerárquicas tienen el inconveniente de que, cuando se realiza una agrupación de clusters, ésta se debe mantener hasta el final no siendo posible una marcha atrás para realizar otro tipo de unión que podría ser más satisfactoria. En la actualidad existen métodos que permiten minimizar el error cuadrático medio entre las similitudes originales y las obtenidas del propio dendrograma, pero suelen ser bastante costosas computacionalmente. Entre los métodos estudiados cabe destacar el algoritmo Branch & Bound de Chandon et al. (1980), el método de programación matemática llevado a cabo por De Soete (1984) - una comparación de ambos métodos puede encontrarse en Chandon & De Soete (1984) -, el método de programación dinámica de Hubert et al. (1998) y los estudios con algoritmos genéticos de Lozano & Larrañaga (1999).

Otro problema que presenta el análisis cluster es que representa de forma más exacta las similitudes mayores, sin embargo, a medida que los clusters crecen en tamaño

las similitudes obtenidas son cada vez menos comparables con las similitudes originales.

A diferencia de los métodos jerárquicos, los métodos no jerárquicos de análisis clúster realizan una clasificación en la que se minimiza la suma de los cuadrados de las distancias entre cada punto y el centroide de su clase. Aquí se pueden emplear diferentes distancias (e.g., euclídea, Chebychev,...) Para aplicar este método hay que predefinir un número arbitrario de clústeres, situar aleatoriamente los centroides de cada clúster, asignar cada punto al centroide más cercano -en función de las distancias definidas, y reevaluar iterativamente las posiciones de los nuevos centroides de cada clúster. No obstante, la mayor dificultad que aparece al aplicar el análisis clúster no jerárquico a la validación de sistemas inteligentes, es la interpretación del concepto de *coordenadas de los puntos*. Esta es una de las razones por las que se prefiere el uso de los métodos jerárquicos de análisis clúster para validar sistemas expertos.

El **escalamiento multidimensional o MDS** (Borg & Groenen, 1997) es una técnica de análisis de datos que permite mostrar a los expertos como puntos en un espacio geométrico, en el que las distancias que separan a los distintos expertos son indirectamente proporcionales a sus similitudes expresadas a través de una determinada medida de pares.

Un algoritmo de MDS apropiado para la validación sería el MDS métrico propuesto por Torgerson (1958) y que se basa en el cálculo de los autovalores y los autovectores de la matriz de distancias, a la cual se le han hecho una serie de transformaciones. Los pasos a seguir en este método son los siguientes:

1. Convertir la matriz de pares inicial en una matriz de disimilitudes (D).
2. Construir una matriz semidefinida positiva A basada en D
3. Obtener las coordenadas de cada elemento a partir de los autovectores y autovalores de la matriz A.

Los resultados del MDS tienden a actuar de forma opuesta a como la hace el análisis cluster, es decir, a representar de forma más exacta las similitudes más pequeñas, cometiendo errores mayores en las similitudes más grandes. Por esta razón suelen ser de mucha utilidad para la interpretación de los resultados los gráficos de burbujas (Figura 1.13). Estos gráficos representan los resultados del MDS en una vista en dos dimensiones superponiendo sobre ellos los resultados del análisis cluster en forma de burbujas.

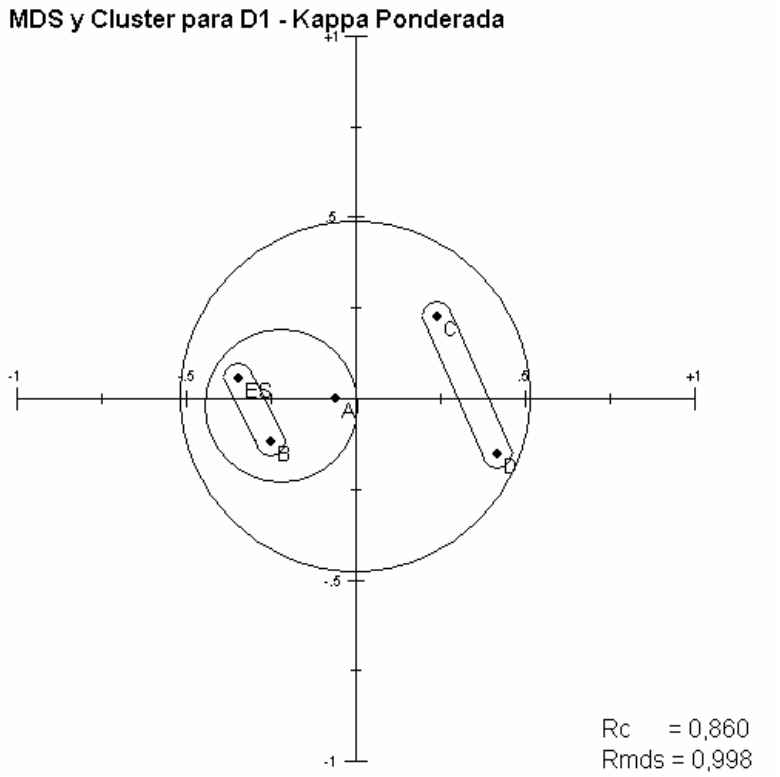


Figura 1.13 Resultados del MDS con los resultados del análisis cluster superpuestos en forma de gráfico de burbujas

Para la validación de los resultados del MDS también puede analizarse el grado de correlación entre las distancias iniciales de los expertos y las extraídas de la gráfica 2D.

Por último citaremos las **medidas de dispersión y tendencia** propuestas originalmente por Duckstein y que fueron utilizadas por Bahill et al. (1995) para validar un sistema de apoyo a la decisión médica.

Esta medida de dispersión trata, como su propio nombre indica, de medir la dispersión de los resultados de un determinado experto en comparación con los resultados del resto de expertos. Su definición se puede ver en la ecuación (1.13) en donde n_c representa el número de casos considerados, n_e el número de expertos y D_{ij} el número de orden del diagnóstico realizado por el experto *i* sobre el caso *j* (las categorías del diagnóstico siguen una escala ordinal a las que se ha asignado un determinado número de orden).

$$\text{Dispersion}_k = \frac{1}{n_c}\sum_{j=1}^{n_c}\left(\sqrt{\frac{1}{n_e - 1}\sum_{i=1}^{n_e}\left(D_{kj} - D_{ij}\right)^2}\right) \tag{1.13}$$

La medida de tendencia es similar a la de dispersión pero trata de mostrar si la magnitud de las interpretaciones de un experto en particular tiende a ser menor o mayor

que la magnitud de los resultados del resto de expertos. Su definición se muestra en la ecuación (1.14).

$$\text{Tendencia}_k = \frac{1}{n_c}\sum_{j=1}^{n_c}\left(\frac{1}{n_e - 1}\sum_{i=1}^{n_e}\left(D_{kj} - D_{ij}\right)\right) \tag{1.14}$$

Las medidas de dispersión y tendencia son las únicas medidas de grupo que no están basadas en un determinado test de pares y se obtienen directamente de la base de datos de validación

1.3.3 Ratios de acuerdo

Los ratios de acuerdo se encargan de medir el acuerdo existente entre un experto (o sistema inteligente) y una referencia estándar. Dicha referencia puede ser un consenso existente entre los expertos, lo que implicaría una validación contra el experto, o la solución real al problema planteado, lo que implicaría una validación contra el problema.

La diferencia fundamental existente entre los tests de pares y los ratios de acuerdo es que los primeros tratan la interpretación en su conjunto, mientras que los segundos analizan los resultados obtenidos en las distintas categorías en las que se divide la interpretación.

Para el cálculo de los ratios de acuerdo se construye una matriz 2×2 para cada una de las categorías que forman una determinada interpretación y se obtienen los ratios de acuerdo para cada una de estas categorías, tal y como se muestra en la Figura 1.14.

		Referencia Estándar		
		D	*¬D*	
Experto	*D*	*a*	*b*	*a + b*
	¬D	*c*	*d*	*c + d*
		a + c	*b + d*	

(a)

Ratios de Acuerdo	
Ratio de verdaderos positivos	a / (a + c)
Ratio de verdaderos negativos	d / (b + d)
Ratio de falsos positivos	b / (b + d)
Ratio de falsos negativos	c / (a + c)
Valor predictivo positivo	a / (a + b)
Valor predictivo negativo	d / (c + d)

Otras medidas de similitud	
Índice de Acuerdo	(a + d) / (a + b + c + d)
Coeficiente de Jaccard	a / (a + b + c)

(b)

Figura 1.14 (a) Tabla de contingencia 2 × 2 para el cálculo de los ratios de acuerdo. D representa la presencia de una decisión o categoría diagnóstica, mientras que ¬D representa su ausencia; (b) ratios de acuerdo y otras medidas de similitud calculadas en base a los datos de la tabla del apartado (a).

Es importante destacar que, además de los ratios de acuerdo, pueden calcularse otras medidas de similitud como el índice de acuerdo o la medida de Jaccard. Esta última es útil en aquellas situaciones en las que se consideran más importantes los desacuerdos positivos que los negativos.

1.4. Metodología de Validación

Como hemos visto en los apartados anteriores la validación no es un proceso sencillo de aplicar. Para facilitar la ejecución de la validación se puede dividir el proceso en tres fases claramente diferenciadas: planificación, aplicación e interpretación.

La **planificación** debe tratar de analizar las características del dominio de aplicación, las características del sistema y las características de la etapa de desarrollo en la que se encuentre el sistema para establecer una serie de estrategias de validación que deben marcar nuestra actuación en las futuras fases.

La **aplicación** tiene como objetivo llevar a la práctica las estrategias establecidas en la fase anterior y consiste básicamente en la aplicación de medidas cuantitativas que pueden darse dentro de un contexto cualitativo.

Dentro de la fase de aplicación es necesario realizar una captura de casuística de validación suficiente y representativa. Generalmente los casos de prueba deberán ser preprocesados para corregir errores, transformar los datos a representaciones más adecuadas e incluir información adicional (como la descripción del formato de la base

de datos, orden de las categorías semánticas, pesos de desacuerdo, etc.). Una vez hemos llevado a cabo la captura y el preprocesado de la casuística se realizan una serie de medidas cuantitativas.

La **interpretación** debe utilizar los resultados de la fase de aplicación para dilucidar si el sistema inteligente se comporta realmente como un experto dentro de su campo de aplicación. Esta fase es la más compleja de la metodología porque los resultados de los tests estadísticos deben tener en cuenta la naturaleza del problema que estamos tratando y las características de la muestra empleada en su obtención.

1.5. Herramientas de Validación

En un reciente estudio, Murrell y Plant (1997) analizaron las principales herramientas de verificación y validación que aparecen en la bibliografía entre los años 1985-1995, de este estudio se desprende que la mayoría de las herramientas descritas realizan tareas de verificación o de refinamiento. El refinamiento es una fase intermedia entre la verificación y la validación que se ocuparía de aplicar pruebas de la "caja blanca" sobre el sistema. El objetivo de estas pruebas es no sólo detectar errores en las interpretaciones del sistema sino también detectar qué estructura de conocimiento ha producido el error. El problema de este tipo de herramientas es que son fuertemente dependientes de la estructura de representación del conocimiento empleada. Entre las distintas herramientas de refinamiento del conocimiento podemos citar KVAT (Mengshoel, 1993), SEEK (Politakis, 1985), SEEK2 (Ginsberg y Weiss, 1985) y DIVER (Zlatareva, 1998).

Otra tipo de herramientas serían aquellas que realizan pruebas denominadas "de la caja negra" porque se centran el los resultados de el sistema inteligente y no en su estructura interna. Esta aproximación tiene la ventaja de que es independiente de la tecnología subyacente en la que se ha implementado el sistema inteligente, de forma que podemos desarrollar una herramienta general que no dependa de la estructura de representación del conocimiento. Otra ventaja que obtenemos es que los índices de rendimiento definidos también son independientes de la implementación y comunes para todo tipo de sistemas inteligentes.

El inconveniente de tratar el sistema inteligente como una caja negra es que los índices obtenidos nos dan una información sobre el rendimiento del sistema pero no nos informan de las causas que provocan errores o problemas (como pasaba en las pruebas de la caja blanca). Sin embargo, aunque la herramienta no indique en qué regla se ha producido el error, sí indica como es el rendimiento del sistema, qué tendencias existen en las interpretaciones, qué categorías semánticas son las más comunes a la salida, etc. Toda esta información sirve al ingeniero del conocimiento para identificar fallos en la estructura del sistema inteligente y para refinar las bases de conocimientos.

Entre las herramientas que siguen la filosofía de la caja negra podemos citar a SHIVA - Sistema Heurístico e Integrado de Validación – (Mosqueira y Moret, 2000) que facilita la realización de cada una de las tareas incluidas dentro de la metodología descrita en el capítulo anterior.

1.6. Resumen

En este capítulo se ha realizado una pequeña introducción a la verificación y validación de los sistemas expertos. La fase de verificación trata de comprobar que el sistema se ha desarrollado correctamente desde tres puntos de vista: verificar el cumplimiento de las especificaciones, verificar los mecanismos de razonamiento y verificar la base de conocimientos. El empleo de shells comerciales y motores de inferencia certificados hace que la importancia de la fase de verificación recaiga fundamentalmente sobre la base de conocimientos. Se han desarrollado muchas herramientas para automatizar la verificación pero su uso sigue teniendo el inconveniente de necesitar que la estructura de la base de conocimientos no sea demasiado compleja.

Una vez que el sistema ha sido verificado debe ser validado. La validación puede verse desde dos perspectivas: una validación orientada a los resultados y una validación orientada al uso (que también suele incluirse dentro de la etapa posterior de evaluación).

Para realizar una validación orientada a los resultados es preciso analizar los distintos aspectos que caracterizan el proceso de validación, entre los que destacamos: personal involucrado, partes del sistema a validar, datos utilizados, criterios de validación, momento en que realizar la validación, errores cometidos y métodos utilizados.

Todas estas características permiten obtener una idea global sobre la problemática que conlleva el proceso de validación de los sistemas expertos. De esta forma podemos dividir el proceso de validación en subfases que simplifiquen su ejecución. Estas fases serán: planificación, aplicación e interpretación.

Las herramientas destinadas a automatizar el proceso de validación se dividen en dos grupos: herramientas de la "caja blanca" y herramientas de la "caja negra" según la visión que tengan de la estructura del sistema que están validando.

1.7. Textos Básicos

- González, Dankel, "Verification and Validation", En: The engineering of knowledge-based systems : theory and practice, Prentice-Hall International, 1993.

- Gupta, U.G. (Ed.) "Validating and Verifying Knowledge-Based Systems", IEEE Computer Society Press, Los Alamitos, California, 1991.

- Gupta U.G. "Validation and Verification of Knowledge-Based Systems: A Survey." Journal of Applied Intelligence, vol. 3, pp. 343-363, 1993.

- Mosqueira-Rey E., Moret-Bonillo, V. "Validation of intelligent systems: a critical study and a tool", Expert Systems with Applications, vol. 18, no. 1, pp. 1-16, 2000.

- O'Keefe, R.M., Balci, O., Smith, E.P. "Validating Expert System Performance." IEEE Expert, vol. 2, no. 4, pp. 81-89, Winter 1987.