

Modelo de Agregación Basado en un Sistema Neurodifuso para un Proceso de Evaluación de Calidad de Software*

Nelly Condori-Fernández
Universidad Nacional de San Agustín
Arequipa, Perú
nelly@unsa.edu.pe

y

Jorge Belenguer Faguás, Manoli Albert Albiol
Universidad Politécnica de Valencia
Departamento de Sistemas Informáticos y Computación
Valencia, España, E46020
{jorbefa,malbert}@dsic.upv.es

Abstract

This paper proposes a new aggregation model for the product quality evaluation process. The evaluation of a complex system supposes a high level of expertise; for this reason, it is insufficient to use only additive scoring models. We propose the use of a neurodifuzzy system to carry out an aggregation process constituted by three levels (attributes, subcharacteristics, characteristics). This system incorporates the learning concepts from the neural networks to the fuzzy inference system. The neural network model known as FAM (Fuzzy Associative Memory) is used for the storage and evaluation of fuzzy rules defined by the expert. Thus, a robust model is obtained which achieves a more precise and objective evaluation.

Keywords: Qualimetric model, ISO 9126, Neural networks, Fuzzy inference system, FAM, Quality Software.

Resumen

Este artículo presenta un nuevo modelo de agregación para el proceso de evaluación de la calidad del producto de software. La evaluación de un sistema complejo implica un nivel alto de experticia, por lo tanto, basarnos sólo en modelos de puntajes aditivos resultan insuficientes. En este trabajo proponemos el uso de un sistema neurodifuso con el objetivo de llevar a cabo un proceso de agregación en tres niveles (atributos, subcaracterísticas y características de calidad). Este sistema incorpora los conceptos de aprendizaje de las redes neuronales a los sistemas de inferencia difuso. El modelo de red neuronal conocido como FAM (Fuzzy Associative Memory) es utilizado para el almacenamiento y evaluación de las reglas difusas definidas por el experto. De esta manera, se obtiene un modelo más robusto logrando una evaluación con un mayor grado de precisión y justificación objetiva.

Palabras claves: Modelo cualimétrico, ISO-9126, Redes neuronales, Sistema de inferencia difusa, FAM, Calidad de software.

* Esta investigación está soportada por el Programa CYTED, en el proyecto VII.18 WEST ; el Proyecto CICYT del programa FEDER, con ref. TIC 1FD97-1102 y de la Sociedad Peruana de Computación

1. Introducción

Durante mucho tiempo, la funcionalidad de un producto ha sido la única manera de medir la calidad del software. Diferentes modelos han sido propuestos y han sido presentados para su uso, como MacCall [13], Boehm [12]. Pero, era necesaria la llegada de un modelo estándar que estableciera las características fundamentales de calidad del software y que ganara el consenso para su aplicación. Es por esta razón, que el comité técnico de la ISO/IEC en 1991, publicó la norma internacional ISO/IEC 9126 [1], la cual establece los lineamientos generales para la evaluación del producto de software a partir de seis características de calidad: funcionalidad, confiabilidad, eficiencia, usabilidad, portabilidad y mantenibilidad.

El alcance de esta norma resulta aún limitado, ya que no contiene directrices sobre métodos y estrategias para todo el proceso de evaluación. A pesar de que la revisión de esta norma sigue siendo un trabajo en progreso. El nuevo estándar ISO/IEC 9126, que se encuentra aún en modo borrador, provee un conjunto de métricas externas[16] e internas[17] que complementa el modelo de calidad descrito en el 9126-1[22]. Sin embargo, para evaluar un sistema como el software no es suficiente identificar un conjunto apropiado de métricas para medir atributos de calidad de software. Puesto que el uso de éstas métricas no elimina la necesidad del juicio humano en un proceso de evaluación. Esto se afirma en la siguiente sentencia: “El uso de métricas de software reduce la subjetividad en la evaluación de la calidad de software al proveer una base cuantitativa para tomar decisiones acerca de la calidad del software” [15].

Por consiguiente, la evaluación de sistemas complejos es primeramente un problema de decisión. Tanto estrategias como técnicas deben contener componentes subjetivos [14]. En el área de la Inteligencia Artificial, algunas técnicas pueden ser aplicadas para modelar esta subjetividad del ser humano. Sin embargo, escasos trabajos se han venido desarrollando al respecto [6], [7], [8]. Es oportuno mencionar que en el trabajo [8], se destacó una desventaja de los sistemas de inferencia difuso, pues a medida que se iba incrementando el número de entradas al sistema (atributos de calidad), el número de reglas difusas se incrementaba dramáticamente, lo que ocasionaba dificultades en el proceso de inferencia.

En este trabajo, nosotros aplicamos la técnica de la lógica difusa con el propósito de facilitar la representación subjetiva de la calidad de software a través de operaciones sobre símbolos o expresiones lingüísticas (bueno, regular, malo, etc.), complementando de esta manera el proceso de cuantificación.

Por otro lado, para el proceso de evaluación de la calidad de software, necesitamos de mecanismos de agregación, que nos permita agrupar convenientemente (atributos, subcaracterísticas, características), hasta producir un resultado global. El modelo Logic Scoring of Preference (LSP)[20],[21], es un modelo útil para la agregación de sistemas complejos. Sin embargo, aún es poco conocido en el área de ingeniería de software.

Las redes neuronales difusas, son el resultado de incorporar características de lógica difusa en modelos neuronales, con capacidad de aprendizaje y procesamiento paralelo. Estas redes neurodifusas pueden ser utilizados con el propósito de optimizar el modelo de agregación LSP.

Esta desventaja que presentan los sistemas de inferencia difuso se ven superados con un sistema de inferencia neurodifuso que combinan las ventajas de las redes neuronales y de la lógica difusa. Logrando obtener un sistema más robusto, confiable y flexible de apoyo al auditor de software.

Por lo tanto, nosotros proponemos un modelo de agregación basado en un sistema neurodifuso, para obtener un sistema más robusto y flexible para el proceso de evaluación de la calidad de software.

En la sección 2, nosotros describimos el marco conceptual de la calidad de software definido por la ISO/IEC 9126. En la sección 3, se presenta un panorama general del proceso de evaluación. En la sección 4, se detalla el modelo de agregación propuesto basado en un sistema neurodifuso donde se explicará la arquitectura de dicho sistema. Finalmente, en la última sección se presentan las conclusiones y trabajos futuros.

2. Marco Conceptual

Conforme al marco conceptual para el modelo de calidad especificado por la norma ISO/IEC 9126-1[19], la calidad del software comprende tres enfoques íntimamente relacionados: proceso, producto y efecto del software.(Vea figura 1)

Con respecto a la evaluación del producto software, se considera tanto la calidad interna como externa del software. Las métricas internas, contribuyen a medir atributos internos (típicamente, medidas estáticas de productos intermedios). Esto permite al usuario detectar problemas y tomar acciones correctivas durante las etapas tempranas de desarrollo del proceso de ciclo de vida [17].

Las métricas externas, (típicamente, medidas del comportamiento del código cuando se está ejecutando), pueden ser solamente utilizadas durante las etapas de prueba del proceso de ciclo de vida y durante algunas etapas operacionales [16]

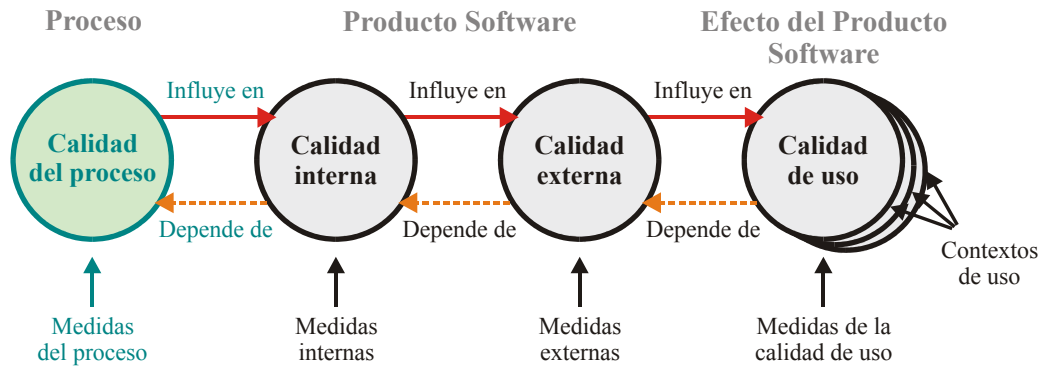


Fig. 1. Marco conceptual para el Modelo de calidad especificado en ISO/IEC 9126-1

De este modo, se requiere de procesos de evaluación de calidad apropiados para que soporten la tarea de medición durante las fases de desarrollo y mantenimiento. El presente trabajo, comprende la evaluación del producto interno de software, por considerarlo prerrequisito indispensable para alcanzar el comportamiento externo requerido. Sin embargo, la esencia del modelo de agregación propuesto no cambia, puede adaptarse fácilmente mediante el ajuste de pesos, y la redefinición de reglas difusas en función a las entradas de una nueva especificación de los criterios de evaluación a considerar.

A continuación se presenta, en primer lugar un panorama acerca de los principales módulos que forman parte del proceso de evaluación propuesto.

3. Panorama general del proceso de evaluación propuesto

El proceso de evaluación propuesto, básicamente está compuesto por tres etapas principales: (Vea figura 2)

- 1.- Especificación de los criterios de evaluación, según la norma ISO/IEC 9126-1.
- 2.- Proceso de Cuantificación, mediante la aplicación de métricas definidas en la norma ISO 9126-3.
- 3.- Proceso de Agregación, basado en un sistema neurodifuso.

En esta sección, se explicará brevemente los dos primeros módulos, el tercer módulo, será explicado en una cuarta sección con mayor detalle, por ser el tema central del artículo.

3.1 Especificación de criterios de evaluación

Es oportuno tener en cuenta lo que menciona Olsina [14], “No se puede medir lo que no se puede comprender” y “Lo que no se puede medir no se puede controlar sistemáticamente”. Por lo tanto, es importante tener presente estos principios, ya que en un proceso de evaluación los evaluadores deben identificar sobre todo los atributos a medir, para que en base a esta especificación se pueda posteriormente analizar, comprender y emplear los resultados apropiadamente.

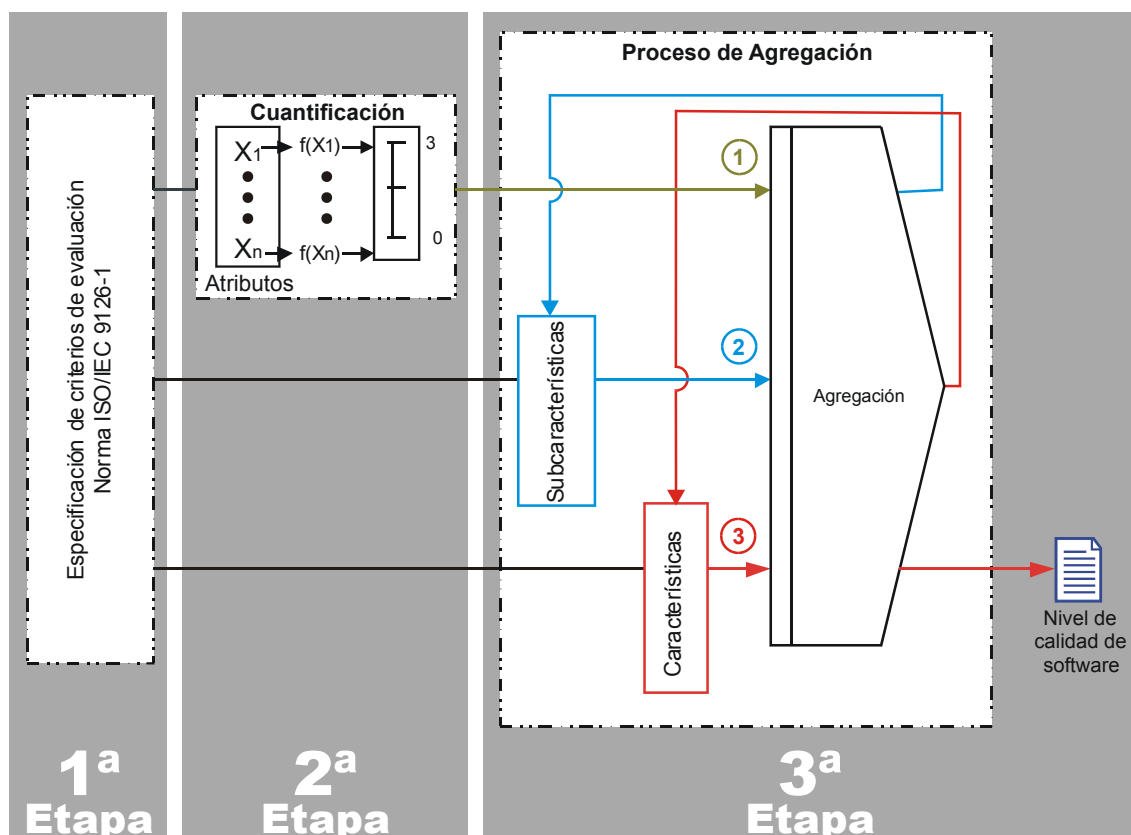


Fig. 2. Esquema general propuesto para el proceso de evaluación de la calidad del software.

Para el presente proceso de evaluación, las características de calidad de más alto nivel, y las sub-características siguen la misma clasificación conceptual de los prescritos en la ISO/IEC 9126-1[19] y con respecto a los atributos, elementos cuantificables, son definidos por la ISO/IEC 9126-3[17]. En base a estos criterios de evaluación se procederá a evaluar la calidad del producto de software. A continuación, se explica el proceso de cuantificación a un nivel de atributos, para posteriormente ser agregados a un nivel de sub-características y características hasta alcanzar un nivel de calidad.

3.2 Proceso de Cuantificación

A partir de la especificación de los criterios de evaluación, para cada atributo cuantificable X_i se debe utilizar una función o métrica ($f(X_i)$) con la finalidad de obtener un valor real entre 0 y 3. Básicamente este proceso consiste en la adaptación de las métricas definidas para cada uno de los atributos internos prescritos en la ISO/IEC 9126-3[17]. En la siguiente tabla sólo se presentan las sub-características y atributos internos de la mantenibilidad, por cuestiones de espacio.

Característica	Subcaracterísticas	Atributos	Fórmula	Interpretación del Valor Medido
Mantenibilidad	Facilidad de Análisis	Registro de Actividades	$F(X)=3(A/B)$ A=Número de ítems a registrar implementados como se especificó en la revisión B=Número de ítems a registrar definidos en las especificaciones	$0 \leq X \leq 3$ Cuanto más cerca de 3 más datos proporcionados para registrar el estado del sistema NOTA: 1. Es necesario convertir este valor al intervalo $<0,3>$ si se hace una sumarización de las características

		Legibilidad de las funciones diagnosticadas	$F(X)=3(A/B)$ A=Número de funciones diagnosticadas implementadas como se especificó en la revisión B=Número de funciones diagnosticadas requeridas	$0 \leq X \leq 3$ Cuanto más cerca de 3 mejor la implementación de las funciones diagnosticadas.
	Modificabilidad	Registrabilidad de modificación	$F(X)=3(A/B)$ A=Número de modificaciones en funciones/módulos con modificaciones comentadas en la revisión B=Número total de modificaciones en las funciones/módulos del código final	$0 \leq X \leq 3$ Cuanto más cerca de 3 más registrable. El control de modificación 0 indica controles de modificación pobres o pequeñas modificaciones, alta estabilidad
	Estabilidad	Impacto de modificación	$F(X)=3(1-A/B)$ A=Número de impactos adversos detectados tras las modificaciones B=Número de modificaciones realizadas	$0 \leq X \leq 3$ Cuanto más cerca de 3 mayor estabilidad.
		Localización del impacto de la modificación	$F(X)=3(A/B)$ A=Número de variables afectadas por la modificación confirmadas en la revisión. B=Número total de variables	$0 \leq X \leq 3$ Cuanto más cerca de 3 mayor impacto de modificación.
	Facilidad de prueba	Complejidad de la función de prueba construida	$F(X)=3(A/B)$ A=Número de funciones de prueba construidas como se especificó en la revisión B=Número de funciones de prueba construidas	$0 \leq X \leq 3$ Cuanto más cerca de 3 más complejidad
		Autonomía de prueba	$F(X)=3(A/B)$ A=Número de dependencias sobre otros sistemas para pruebas que han sido simuladas con matrices B= Número total de dependencias de prueba sobre otros sistemas	$0 \leq X \leq 3$ Cuanto más cerca de 3 mejor
		Observabilidad del proceso de prueba	$F(X)=3(A/B)$ A=Número de puntos de control especificados en la revisión B=Número de puntos de control diseñados	$0 \leq X \leq 3$ Cuanto más cerca de 3 peor.
	Conformidad	Conformidad de mantenibilidad	$F(X)=3(A/B)$ A= Número de ítems correctamente implementados relacionados con la conformidad de mantenibilidad confirmados en la evaluación B= Número total de ítems de conformidad	$0 \leq X \leq 3$ Cuanto más cerca de 3 más conformidad

Tabla 1. Relación de métricas para el conjunto de atributos definidos para la mantenibilidad. (Tomado del estándar de la ISO 9126-3)

4. Modelo de agregación basado en un sistema neurodifuso

En esta sección, se presenta un modelo de agregación a tres niveles, como se observa en la figura 2. Cada uno de los valores cuantificados de los atributos (segunda etapa), pasa por un proceso de agregación (tercera etapa) con la finalidad de obtener valores representativos a un nivel de sub-características. Luego, estos valores pasan nuevamente por un

proceso de agregación para obtener los valores representativos a un nivel de características. Finalmente, por tercera y última vez dichos valores son agregados obteniéndose un nivel de calidad de software.

La arquitectura básica propuesta para este proceso de agregación, consiste en un sistema de inferencia difuso capaz de tratar variables numéricas con variables lingüísticas de modo formal [5], el mismo que comprende de cinco componentes funcionales como se presenta en la figura 3. El motor de inferencia de dicho sistema, está conformado por un conjunto de redes neuronales asociativas para almacenar cada una de las reglas difusas previamente definidas.

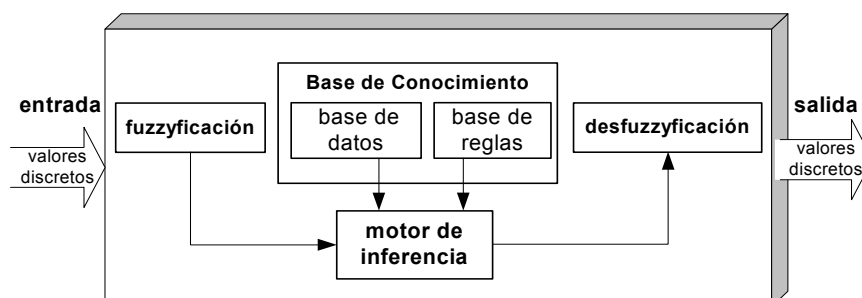


Fig. 3. Sistema de Inferencia Difuso

Por lo tanto, el principal objetivo de los sistemas neurodifusos es combinar los beneficios de ambas técnicas, permitiendo que los sistemas sean más robustos, y faciliten la integración del conocimiento cualitativo y cuantitativo [2], [10]. Este proceso de inferencia es explicado con mayor detalle mediante un ejemplo hipotético a través de cada uno de los componentes de dicho sistema.

4.1 Fuzzyficación¹

Para llevar a cabo dicho proceso, es necesario definir variables lingüísticas, valores lingüísticos, funciones de pertenencia y un dominio de valores discretos [10]. Por motivos de espacio, sólo se presenta los elementos necesarios a un nivel de características. Sin embargo, para el resto de variables lingüísticas correspondientes a niveles de sub-características y atributos los valores lingüísticos, las funciones de pertenencia y el dominio de valores permanecen igual.

	<i>VARIABLES lingüísticas</i>	<i>Id</i>	<i>Valores lingüísticos</i>	<i>Funciones de pertenencia</i>	<i>Dominio de valores discretos</i>			
Antecedente	Funcionalidad	X1	Malo, Regular, Bueno	Forma triangular y trapezoidal	<i>Bueno</i>	<i>Regular</i>	<i>Malo</i>	
	Confiabilidad	X2			[0, 1.5]	[0.5, 2.5]	[1.5, 3]	
Consecuente	Nivel de calidad	Y	Malo, Regular, Bueno, Muy Bueno	Forma triangular y trapezoidal	<i>Malo</i>	<i>Regular</i>	<i>Bueno</i>	<i>Muy Bueno</i>
					[0, 1.5]	[0.5, 2]	[1.5, 2.5]	[2, 3]

Table 2. Elementos principales para el proceso de fuzzyficación en un tercer nivel

La Figura 4, representa de manera gráfica las funciones de pertenencia, tanto para el antecedente como para el consecuente. (Vea figura 4)

¹ El término fuzzyficación es empleado para referirse al proceso por el cual las variables reales son transformadas a variables difusas. Del mismo modo, el término “desfuzzyficación” se empleará para referirse al proceso contrario, es decir al proceso por el cual las variable difusas son transformadas en variables reales.

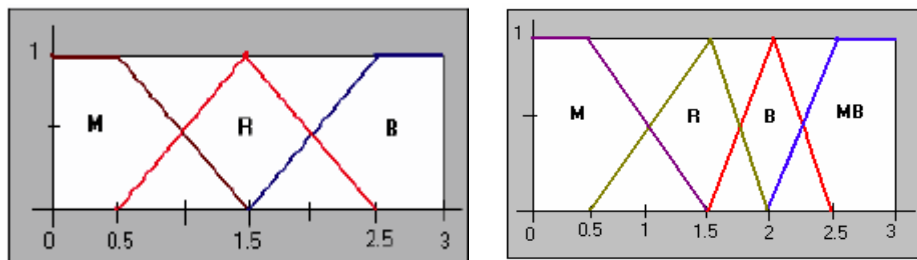
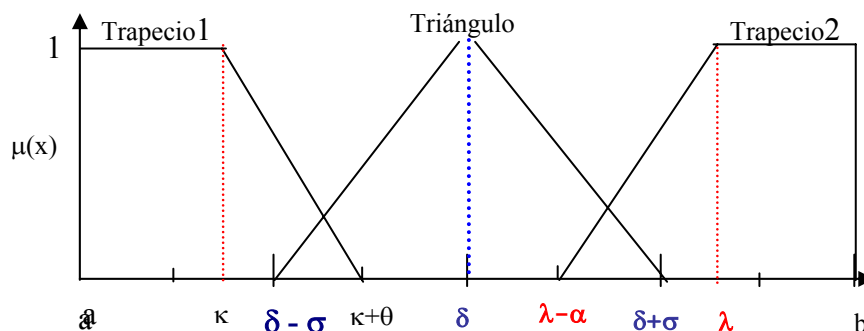


Fig. 4. Funciones de pertenencia del Antecedente (Izquierda) y Consecuente (Derecha) de una Regla Difusa



A continuación, se presentan de manera genérica las ecuaciones paramétricas respectivas en función de los valores que determinan la altura de los conjuntos difusos (k, δ, λ) y valores de variación (σ, θ, α), tanto para la forma triangular como trapezoidal:

Para el caso del triángulo:

$$\mu_A = \begin{cases} (1/\sigma)x - (1/\sigma)(\delta - \sigma) & (\delta - \sigma) \leq x < \delta \\ 1 & x = \delta \\ (1/\sigma)(\delta + \sigma) - (1/\sigma)x & \delta < x \leq (\delta + \sigma) \end{cases}$$

Para el caso del trapecio1:

$$\mu_B = \begin{cases} 1 & a \leq x \leq \kappa \\ (1/\theta)(\kappa + \theta) - (1/\theta)x & \kappa < x \leq (\kappa + \theta) \end{cases}$$

Para el caso del trapecio2:

$$\mu_C = \begin{cases} \frac{1}{\alpha}x - \frac{1}{\alpha}(\lambda - \alpha) & \lambda - \alpha \leq x \leq \lambda \\ 1 & \lambda < x \leq b \end{cases}$$

Con la finalidad de simplificar la explicación del proceso de fuzzyficación se ejemplificará al tercer nivel de evaluación (características), debido a que el procedimiento para el primero y segundo nivel es semejante. Para ello se presenta la siguiente tabla, donde se muestra la simbología para la representación de las variables y valores lingüísticos así como de los valores reales a fuzzyficar.

Variables Lingüísticas	Mala	Regular	Buena	Muy Buena	Valor Real
Funcionalidad (x1)	A1	A2	A3		2.7
Confiabilidad (x2)	B1	B2	B3		2.4
Mantenibilidad (x3)	C1	C2	C3		2.2
Calidad (y)	Q1	Q2	Q3	Q4	

Table 3. Notación para las Variables y valores lingüísticos

En base a la figura 4 (anterior) y las ecuaciones paramétricas generales que se presentaron anteriormente, obtuvimos los siguientes valores difusos para cada uno de los conjuntos (malos, regulares, buenos) de cada variable lingüística:

$a = 0$	Funcionalidad: Sea $x = 2.7$	Confiabilidad: Sea $x = 2.4$	Mantenibilidad: Sea $x = 2.2$
$k = 0.5$ $\theta = 1$	$\mu_{A1} = 0$	$\mu_{B1} = 0$	$\mu_{C1} = 0$
$\sigma = 1$ $\delta = 1.5$	$\mu_{A2} = 0$	$\mu_{B2} = 0.1$	$\mu_{C2} = 0.3$
$\alpha = 1$ $\lambda = 2.5$	$\mu_{A3} = 1$	$\mu_{B3} = 0.9$	$\mu_{C3} = 0.7$
$b = 3$			

4.2 Base de Conocimiento

Como se observa en la figura 3, la base de conocimientos comprende de la base de datos y la base de reglas. Los conjuntos difusos, explicados anteriormente son definidos en la base de datos y las reglas difusas en la base de reglas. Cada regla definida está compuesta por tres antecedentes, conectados por el operador lógico “AND”.

Como se puede observar en la tabla 3, las tres variables de entrada constan de tres términos lingüísticos cada una. El sistema tiene entonces: $3^3 = 27$ posibles combinaciones. De estos 27 estados, se detectaron 13 estados imposibles. La eliminación de estos estados se basó en la relación directa que existe entre funcionalidad-confiabilidad y la repercusión que tienen ambas sobre la mantenibilidad. Por ejemplo, resulta contradictorio hablar de un producto con una mala funcionalidad y al mismo tiempo que este producto tenga una buena confiabilidad. Por lo tanto, nuestra base de reglas está conformada por 14 reglas difusas. Según el ejemplo que se tomó en la fase de fuzzyficación, las reglas activadas de la base de conocimientos son las enmarcadas por la línea roja. (Vea tabla 4).

Base de Reglas Difusas				
Nro	Antecedente	Consecuente		
1	R1: Si x1=A1 AND x2= B1 AND x3 = C 1	ENTONCES y = Q1		
2	R2: Si x1=A1 AND x2= B1 AND x3 =C2	ENTONCES y =Q1		
	R3: Si x1=A1 AND x2= B1 AND x3 =C3	ENTONCES y =Q2		
	R4: Si x1=A1 AND x2= B2 AND x3 =C1	ENTONCES y =Q1		
	R5: Si x1=A1 AND x2= B2 AND x3 =C2	ENTONCES y =Q2		
	R6: Si x1=A1 AND x2= B2 AND x3 =C3	ENTONCES y =Q2		
	R7: Si x1=A1 AND x2= B3 AND x3 =C1	ENTONCES y =Q2		
	R8: Si x1=A1 AND x2= B3 AND x3 =C2	ENTONCES y =Q2		
	R9: Si x1=A1 AND x2= B3 AND x3 =C3	ENTONCES y =Q3		
3	R10 Si x1=A2 AND x2= B1 AND x3 =C 1	ENTONCES y =Q1		
4	R11 Si x1=A2 AND x2= B1 AND x3 =C2	ENTONCES y =Q2		
	R12 Si x1=A2 AND x2= B1 AND x3 =C3	ENTONCES y =Q2		
5	R13 Si x1=A2 AND x2= B2 AND x3 =C 1	ENTONCES y =Q2		
6	R14 Si x1=A2 AND x2= B2 AND x3 =C2	ENTONCES y =Q2		
	R15 Si x1=A2 AND x2= B2 AND x3 =C3	ENTONCES y =Q3		
7	R16 Si x1=A2 AND x2= B3 AND x3 =C 1	ENTONCES y =Q2		
8	R17 Si x1=A2 AND x2= B3 AND x3 =C2	ENTONCES y =Q2		
9	R18 Si x1=A2 AND x2= B3 AND x3 =C3	ENTONCES y =Q4		
	R19 Si x1=A3 AND x2= B1 AND x3 =C 1	ENTONCES y =Q1		
	R20 Si x1=A3 AND x2= B1 AND x3 =C2	ENTONCES y =Q2		
	R21 Si x1=A3 AND x2= B1 AND x3 =C3	ENTONCES y =Q3		
10	R22 Si x1=A3 AND x2= B2 AND x3 =C 1	ENTONCES y =Q1		
11	R23 Si x1=A3 AND x2= B2 AND x3 =C2	ENTONCES y =Q3		
12	R24 Si x1=A3 AND x2= B2 AND x3 =C3	ENTONCES y =Q4		
	R25 Si x1=A3 AND x2= B3 AND x3 =C 1	ENTONCES y =Q3		
13	R26 Si x1=A3 AND x2= B3 AND x3 =C2	ENTONCES y =Q4		
14	R27 Si x1=A3 AND x2= B3 AND x3 =C3	ENTONCES y =Q4		

Estado imposible

Estado posible

Tabla 4. Conjunto de reglas difusas

4.3 Motor de Inferencia

Además de determinar una base de reglas difusas, el *motor de inferencia* es otro componente fundamental. El modelo de red neuronal difusa conocida como FAM (Fuzzy Associative Memory), propuesto por B. Kosko[4], es utilizado como memoria asociativa para el almacenamiento y evaluación de las reglas difusas. Dicho motor de inferencia está formado por 14 redes FAM_i que representan el número de reglas difusas determinadas. Cada una de estas redes está conformada por 3 subredes FAM_{AiQj} que representa el número de antecedentes por regla. Finalmente, cada subred está compuesta por 7 neuronas en la capa de entrada para las variables lingüísticas (x1, x2, x3) y 11 neuronas en la capa de salida para la variable lingüística “y”. Estos valores representan el número de valores discretos a tomarse en cuenta y fueron determinadas empíricamente por pruebas de ensayo y error.

En la figura 5 con fines de reducir la complejidad de la misma, sólo se muestra el esquema del motor de inferencia para las reglas difusas activadas en base a los valores difusos obtenidos para cada una de las variables lingüísticas (x1, x2, x3) durante la fuzzyficación.

Para la evaluación de las reglas difusas, es necesario considerar 2 aspectos:

- 1.- *Ponderar las reglas* que van a gobernar el funcionamiento del sistema. El valor de los pesos para todas las reglas es de 1 por considerarse a todas de igual importancia.
- 2.- *Adicionar una última capa* de 11 neuronas equivalente al número de neuronas de la capa de salida de una subred FAM. Como se muestra en la figura 5, los valores de los subconjuntos difusos Qi extraídos de cada una de las redes FAM_i forman la entrada de esta capa encargada de realizar la suma de dichos valores para obtener un conjunto global difuso, que represente el nivel de calidad del software.

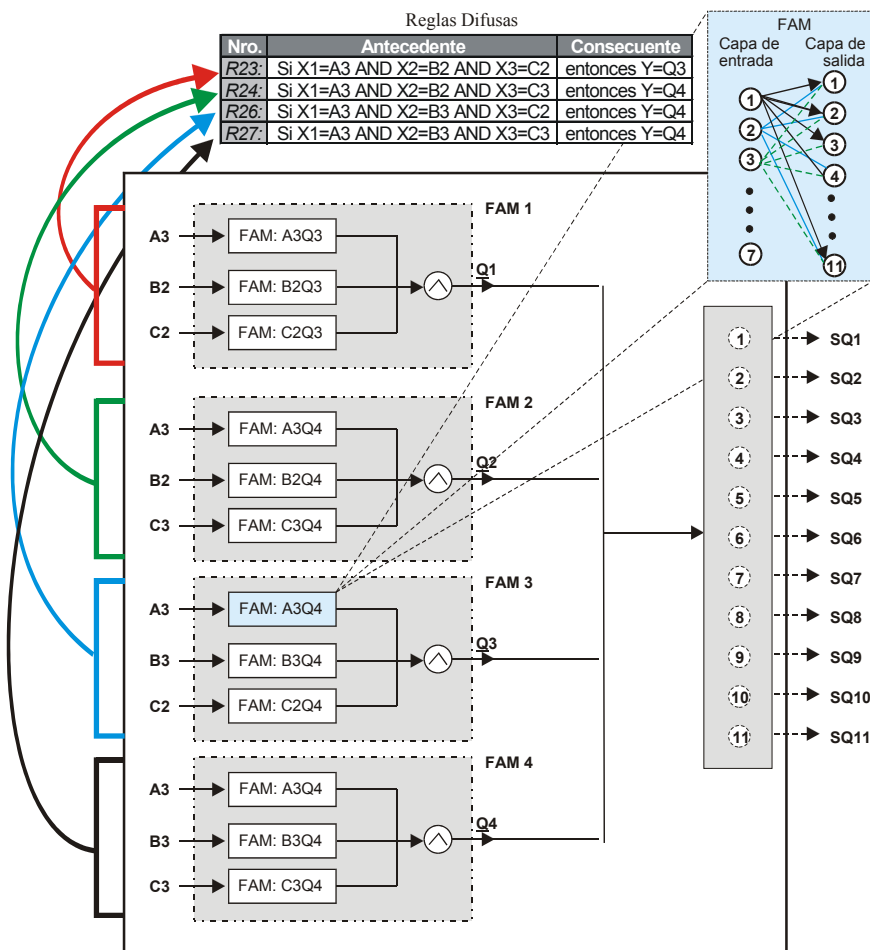


Fig. 5. Esquema del motor de inferencia para el proceso de agregación

En base a la estructura definida para la evaluación de las reglas difusas, podemos expresar matemáticamente la fase de aprendizaje y funcionamiento de una FAM [4],[11] de la siguiente forma: (Tabla 5)

FASES	FAM :A1Q1	FAM :B1Q1	FAM :C1Q1
Aprendizaje	$W_{A3Q3} = A_3^T \circ Q_3$	$W_{B3Q3} = B_2^T \circ Q_3$	$W_{C1Q1} = C_2^T \circ Q_3$
Funcionamiento	$r_1 = E_A \circ W_{A3Q3}$	$r_2 = E_B \circ W_{B2Q3}$	$r_3 = E_C \circ W_{C2Q3}$

Table 5. Aprendizaje y funcionamiento del sistema de inferencia difuso

Como la escala de valores reales es la misma y las funciones de membresía se mantienen constantes. Los valores discretos que representan al número de neuronas de la capa de entrada son:

$$A_i = \{0.3, 0.7, 1, 1.5, 2.8, 2.5, 2\}$$

Y los discretos que representan al número de neuronas de la capa de salida son:

$$Q_i = \{0.2, 0.5, 0.7, 1, 1.3, 1.5, 1.7, 2, 2.3, 2.6, 3\}$$

Para la salida Q_1 , se tiene: $Q_1 = r_1 \wedge r_2 \wedge r_3$ es decir: $Q_1 = \min\{r_1, r_2, r_3\}$

De igual manera es para: Q_2, Q_3 y Q_4 .

Para la salida Total del sistema S_Q , se tiene: $S_Q = Q_1 p_1 + Q_2 p_2 + Q_3 p_3 + Q_4 p_4$

Donde p_i representa el valor de los pesos para cada una de las reglas definidas.

En cuanto a los vectores de entrada para la red (E_A, E_B y E_C) se obtienen de la siguiente manera: Es 1 para el valor “más cercano” al valor de la entrada y las demás componentes tienen el valor de 0. Este concepto de “más cercano” implica pasar por el módulo de fuzzyficación para obtener valores de pertenencia a un determinado conjunto difuso.

4.4 Desfuzzyficación

Existen una variedad de métodos para el proceso de desfuzzyficación, el método seleccionado es el del centro _ máximo [9], que resulta de la combinación de dos métodos, conocidos como el método del máximo y el método del centroide. (Ecuación 7)

$$y = \frac{\sum_{i=1}^7 A_i(x) * a_i}{\sum_{i=1}^7 A_i(x)}$$

Donde:

$A_i(x)$es el grado de pertenencia (altura) de la muestra i-ésima del rango de valores discretos.

a_ies el valor del dominio donde se encuentra el pico de la función de pertenencia.

Por lo tanto, el valor desfuzzyficado representa el nivel de calidad del software para la etapa de mantenimiento. Dicho valor es almacenado en una base de datos para facilitar el seguimiento en la calidad del software.

5. Conclusiones y Trabajos futuros

En el artículo se ha presentado un modelo de agregación basado en un sistema de inferencia neurodifuso. Se hace uso de la técnica de la lógica difusa con el propósito de facilitar la representación subjetiva de la calidad de software, complementando de esta manera el proceso de cuantificación. El modelo propuesto incorpora conceptos de aprendizaje de redes neuronales a los sistemas de inferencia difusa, ya que dichos sistemas carecen de la adaptabilidad necesaria para reaccionar tras una modificación de alguna de las condiciones externas, como algún cambio en los criterios de evaluación.

En el artículo se ha hecho uso del modelo de calidad especificado por el nuevo estándar ISO/IEC 9126. Aunque dicha norma está en fase de borrador, nuestra propuesta es independiente del modelo de calidad a emplear, pudiendo ser fácilmente adaptada, mediante el ajuste de pesos y la redefinición de reglas difusas, a una nueva especificación de los criterios de evolución a considerar.

El principal logro de éste trabajo radica en la innovación que supone aplicar conceptos provenientes del área de la Inteligencia Artificial, tales como las redes neuronales difusas, a procesos de evaluación de la calidad del software. Hasta el momento de la publicación del presente trabajo, los autores no han encontrado en la literatura ninguna aproximación similar que plantee un proceso de evaluación de la calidad del software enriquecido mediante inferencia neurodifusa.

En la actualidad se está estudiando la posibilidad de desarrollar una herramienta inteligente que dé soporte al auditor de software, con la finalidad de mejorar la calidad de sus procesos. Del mismo modo, como trabajo futuro también se pretende afinar de manera más precisa el motor de inferencia mediante una aproximación formal del número de neuronas en la capa de entrada y salida. Así como mejorar el ajuste y refinado de las funciones de pertenencia a los conjuntos difusos definidos para la evaluación de la calidad del software.

Referencias

- [1] ISO/IEC 9126:1991. Information technology - Software product evaluation - Quality characteristics and guidelines for their use.
- [2] Espinoza J., Vandewalle J., "Técnicas neurodifusas. Una alternativa aplicable al control automático", 1995.
- [3] Martin J.D., "Implementación de redes neurodifusas para ser aplicadas en problemas de clasificación y modelización", USA 2000, <http://www.dissertation.com/lybrary/112113xa.htm>.
- [4] Hilera J., "Redes Neuronales Artificiales, Fundamentos, Modelos y Aplicaciones", Addison-Wesley Iberoamericana. 1995, pp 322-368.
- [5] Jyh-Shing R. J., "Adaptive-Network-Based Fuzzy Inference System" IEEE Transactions on systems, Man and Cybernetics, vol. 23, pp 665-685.
- [6] Galvão L. "Um Modelo de Avaliação de Especificações Semi-Formais de Requisitos de Software Baseado na Teoria dos Conjuntos Nebulosos", Actas IV Workshop en Requisitos, Buenos Aires 2001.
- [7] Condori N., "Sistema Híbrido Neurodifuso para la Evaluación de la Calidad del Software en un Proceso de Auditoria Informática", Actas de las Jornadas Chilenas de Ciencias de la Computación II Workshop de Inteligencia Artificial, Punta Arenas, Noviembre-2001.
- [8] Condori N., "Modelo de Evaluación de Especificación de Requisitos de Software Basado en un Sistema de Inferencia Difuso". Memorias 5to Workshop de Ingeniería de Requisitos y Desarrollo de Ambientes Software, IDEAS, La Habana-Cuba, Abril-2002. ISBN: 959-7160-14-5
- [9] Timothy J.R., "Fuzzy Logic with Engineering Applications", McGraw-Hill, 1995, pp 130-147.
- [10] Ojala T., "Neuro-Fuzzy Systems In Control", Master Thesis, Electrical Engineering, Tampere, Finland, 1994.
- [11] Valluru R. Y Hayagriva R., "C++ Neural Networks and Fuzzy Logic", Miss: Press, 1995, pp 218-240.
- [12] Boehm, B.; Brown, J.R.; Kaspar, J.R., et al. 1978, "*Characteristics of Software Quality*", TRW Series of Software Technology.
- [13] McCall, J.A; Richards, P.K.; Walters, G.F.; 1977, "*Factors in Software Quality*", RADC TR-77-369.
- [14] Olsina, L. 1999, Tesis: "Metodología Cuantitativa para la Evaluación y Comparación de la Calidad de Sitios Web", La Plata-Argentina.
- [15] IEEE Std 1061-1992, "*IEEE Standard for a Software Quality Metrics Methodology*", IEEE Computer Society Press
- [16] ISO/IEC 9126-2 International Standard, 1999 (Draft) "Information technology – Software product evaluation – Part 2: *External Metrics*".
- [17] ISO/IEC 9126-3 International Standard, 1999 (Draft) "Information technology – Software product evaluation – Part 3: *Internal Metrics*".
- [18].Zadeh L.A., "The concept of a linguistic variable and its Application to Approximate Reasoning", Information Science, Part I.
- [19] ISO/IEC 9126-1 International Standard (Draft), 1998 "Information technology – Software product evaluation – Part 1: *Quality Model*".
- [20] Dujmovic, J.J., "Preferential Neural Network. Chapter 7 in Neural Network – Concepts, Applications, and Implementations", Edited by P. Antognetti and V. Milutinovic. Prentice Hall Advanced Reference Series, Vol. 2, pp.155-206. 1991
- [21] Dujmovic, J.J., "A Method for Evaluation and Selection of Complex Hardware and Software Systems", The 22nd International Conference for the Resource Management and Performance. 1996