

Bases de Datos Difusas Modeladas con UML

Marcela Varas Contreras

Universidad de Concepción, Depto. Ingeniería Informática y Ciencias de la Computación,
Concepción, Chile
mvaras@udec.cl

y

Angélica Urrutia Sepúlveda

Universidad Católica del Maule, Departamento de Computación e Informática
Talca, Chile
aurrutia@hualo.ucm.cl

Resumen

En este artículo se presenta un diseño de una base de datos difusa en UML. Se sentaron las bases conceptuales para la definición de atributos difusos y de los mecanismos que permiten consultar la base de datos difusa una vez creada, esto es mediante el uso de comparadores difusos. Se utilizaron los conceptos de estereotipo y patrón provistos por UML para formalizar una propuesta que queda a disposición de los diseñadores.

Palabras claves: Bases de Datos Difusas, Atributos Difusos en UML, Datos Imprecisos, Modelo de Datos Difusos.

1 Introducción

El proceso del diseño de bases de datos se desarrolla en tres fases: diseño conceptual, diseño lógico y diseño físico. Este trabajo se centra en la primera fase, el diseño conceptual, por su característica de analizar los requisitos que debe tener una base de datos desde un punto de vista independiente de la tecnología a utilizar en su implementación. Un modelo de datos de un sistema de software está especificado en un lenguaje de modelado (UML, EER entre otros). Cada modelo tiene semántica y notación, éste puede adoptar varios formatos que incluyen texto y gráficos. Rumbaugh et. al. [7] afirman que un propósito fundamental de los modelos es que permiten “captar y enumerar exhaustivamente los requisitos y el dominio de conocimiento, de forma que todos los implicados puedan entenderlos y estar de acuerdo con ellos”. Es por ello, que este trabajo se centra en la captura de aquellos datos que posean un dominio impreciso que actualmente no son representados por los modelos, debido a la falta de expresividad de los lenguajes de modelación, nuestra propuesta utiliza la herramienta de modelado UML.

UML permite ser extendido para aumentar su expresividad. Utilizamos esta característica más el diseño de un patrón para proveer a los diseñadores la posibilidad de incluir atributos difusos y aprovechar sus características.

Algunas propuestas en esta línea se encuentran en Marín et. al. [6], que incorporan la vaguedad para definir el concepto de tipos difusos, así como también de la instanciación y la herencia aplicada a dicho concepto, pero todo ello a nivel de programación. Una investigación que profundiza en el modelado conceptual de datos es Ma et al. [5], quienes proponen un tipo de atributo que indica el grado de importancia que éste tiene para cada entidad o clase. Para el caso de nuestra propuesta usamos tipos de atributos que poseen dominios imprecisos o difusos, los cuales se encuentran definidos en una clase. Por otro lado, Geneste y Ruet [1], [2] proponen una clase que incorpora un trapezoide de una etiqueta lingüística tratado como grado de pertenencia en una aplicación específica de un caso tratado en UML. Esta propuesta es simplista, pues normalmente, como se muestra en la sección 2, los atributos difusos tipo 2 obedecen a más de una etiqueta lingüística. Nuestro propósito es desarrollar un modelo genérico que sirva para cualquier especificación de requerimientos del usuario con datos imprecisos.

Originalmente, el nivel conceptual permite la utilización de tipos de datos elementales a los que se les llama clásicos (crips). Estos tipos de datos son numéricos, alfanuméricos y datos binarios. Existen otro tipo de datos, que no son tratados en los modelos de datos conceptuales clásicos o tradicionales, que contienen incertidumbre o imprecisión en su información. Estos últimos se consideran datos “difusos” (fuzzy) asociados a la teoría de conjuntos difusos que los humanos manejamos de forma cotidiana y natural ([10]). Por concepto o información difusa entendemos información que encierra alguna imprecisión o incertidumbre.

Este artículo presenta una serie de conceptos básicos de la teoría de conjuntos difusos y atributos difusos en la sección 2, con una introducción a patrones y estereotipos con UML, elementos que no son tan ampliamente difundidos como los conceptos elementales de la vista estática de UML. La sección 3 presenta las bases de nuestra propuesta, para en la sección 4 ejemplificar su uso con el desarrollo de un caso sencillo, pero generalizable a casos más complejos. Cerramos el artículo con las conclusiones y trabajo futuro.

2 Conceptos Básicos

UML (Lenguaje Unificado de Modelado), es un lenguaje de modelado visual que se usa para especificar, visualizar, construir y documentar artefactos de un sistema de software. Captura decisiones y conocimiento sobre los sistemas que se deben construir, su objetivo es lograr que,

además de describir con cierto grado de formalismo tales sistemas, puedan ser entendidos por los usuarios de aquello que se modela. Nuestra investigación hace uso de las herramientas de UML para el modelado de datos, especialmente se basará en estereotipos y patrones, como una forma especial de representar atributos con imprecisión utilizando la teoría de conjuntos difusos (lógica difusa).

Estereotipo en UML: Es un nuevo elemento del lenguaje definido sobre la base de algún elemento pre existente de UML. Extienden la semántica pero no la estructura de las clases del metamodelo. Permite representar una variación de un elemento existente que posee otra intención, o distinción de uso. La definición de un estereotipo se hace en forma explícita en la vista estática, mediante una relación de generalización con el elemento de UML que es base para su definición. El nombre del estereotipo debe ser distinto de los elementos de UML, y se denota entre comillas francesas («nombre estereotipo»). También puede considerarse una notación gráfica distintiva.

Patrones: Solución ya probada y eficaz para algún problema de diseño que puede expresarse como un conjunto de principios y heurísticas[4].

Conjuntos difusos: La teoría de conjuntos difusos (también llamada borrosos) parte de la teoría clásica de conjuntos, añadiendo una función de pertenencia al conjunto, definida ésta como un número real entre 0 y 1. Así, se introduce el concepto de conjunto o subconjunto difuso asociado a un determinado valor lingüístico, definido por una palabra, adjetivo o etiqueta lingüística A . Para cada conjunto o subconjunto difuso se define una *función de pertenencia o inclusión* $\mu_A(u)$, que indica el grado en que la variable u está incluida en el concepto representado por la etiqueta ([10]). Un conjunto difuso A sobre un universo de discurso U (dominio ordenado) es un conjunto de pares dado por:

$$A = \{\mu_A(u) / u : u \in U, \mu_A(u) \in [0,1]\}$$

Donde, μ es la llamada función de pertenencia y $\mu_A(u)$ es el *grado de pertenencia* del elemento u al conjunto difuso A . Este grado oscila entre los extremos 0 y 1, definido como:

$\mu_A(u) = 0$, indica que u no pertenece en absoluto al conjunto difuso A .

$\mu_A(u) = 1$, indica que u pertenece totalmente al conjunto difuso A .

Por ejemplo si consideramos el valor lingüístico *estatura_de_una_persona* podría definirse tres subconjuntos difusos, cada uno identificado por una etiqueta {bajo, medio, alto}, y con una función de pertenencia $\mu_{bajo}(u)$, $\mu_{medio}(u)$, $\mu_{alto}(u)$, respectivamente.

Por otro lado, para dominios con referencial no ordenado tenemos la *función de similitud*. Esta función define que, para cada dominio D , escalar o numérico, se establece una relación de similitud que sirve para medir la similitud o parecido entre cada dos elementos del dominio. Normalmente, los valores de similitud están normalizados en un intervalo $[0,1]$, correspondiendo el 0 al significado “totalmente diferente” y el 1 al significado “totalmente parecido” o iguales ([3]). Por tanto, una relación de similitud puede ser vista como una función s_r , tal que:

$$s_r : D \times D \rightarrow [0,1]$$

$$s_r(d_i, d_j) \mapsto [0,1] \text{ con } d_i, d_j \in U$$

Tanto los dominios con referencial ordenado y no ordenado pueden representar de forma adecuada conceptos de “imprecisión” con la teoría de conjuntos difusos. Es necesario hacer notar que muchos de estos conceptos naturales dependen, en mayor o menor medida, de la persona que los expresa.

Atributos difusos: Para modelar atributos difusos consideraremos datos con dominio de

referencial ordenado y no ordenado. En un referencial ordenado tenemos las representaciones de: Distribución de posibilidad trapezoidal, Etiquetas lingüísticas, Valores aproximados, Intervalos de posibilidad. En un referencial no ordenado: Escalares simples y Distribución de posibilidad sobre escalares. También son considerados los valores Unknown, Undefined y Null ([3]). Por otro lado, los atributos clásicos que su dominio sea impreciso pueden ser tratados pero con un formato especial para representar atributos difusos. La clasificación adoptada se basa en criterios de representación y de tratamiento de los datos “imprecisos”: se clasifica según el tipo del dominio que les subyace y por si permiten representar la información imprecisa o sólo permiten el tratamiento impreciso de datos sin imprecisión ([6]). Los *atributos difusos* pueden ser de 3 tipos:

- **Tipo 1:** Estos son atributos con “*datos precisos*”, clásicos o *crisp* (tradicionales, sin imprecisión), que pueden tener etiquetas lingüísticas definidas sobre ellos. Este tipo de atributos reciben una representación igual que los datos precisos, pero admiten que se puedan transformar utilizando condiciones difusas.
- **Tipo 2:** Son atributos que pueden recoger “*datos imprecisos sobre referencial ordenado*”. Estos atributos admiten tanto datos *crisp* como difusos, en forma de distribuciones de posibilidad sobre un dominio subyacente ordenado.
- **Tipo 3:** Son atributos sobre “*datos de domino discreto no ordenado con analogía*”. En estos atributos se definen algunas etiquetas (“rubio”, “pelirrojo”, “castaño...”), que son escalares con una *relación de similitud* (o proximidad) definida sobre ellas, de forma que esta relación indique en qué medida se parecen entre sí cada par de etiquetas.

Hay que considerar que, el conjunto difuso de posibles valores que puede tomar una determinada característica se denomina *dominio difuso*. Cada una de las propiedades o características que tiene una clase se denomina atributo.

Comparadores Difusos.

Además de los comparadores comunes (=,<,>, etc.), un FSQL incluye los comparadores difusos de las tablas 1 y 2 [3]. Al igual que el SQL los comparadores difusos pueden comparar dos columnas del mismo tipo o de tipos compatibles. Éstos pueden ser clasificados en comparadores de posibilidad o necesidad.

Los *comparadores de posibilidad* son más generales (menos restrictivos) que los de necesidad. Por tanto, los *comparadores de necesidad* recuperan menos instancias y estas instancias cumplirán necesariamente con las condiciones impuestas en la consulta.

POSIBILIDAD

Comp. Difuso	Significado
FEQ	Posiblemente Igual
FGT	Posiblemente Mayor
FGEQ	Posiblemente Mayor o Igual
FLT	Posiblemente Menor
FLEQ	Posiblemente Menor o Igual
MGT	Posiblemente Mucho Mayor
MLT	Posiblemente Mucho Menor

Tabla 1. Comparadores de posibilidad.

NECESIDAD

Comp. Difuso	Significado
NFEQ	Necesariamente Igual
NFGT	Necesariamente Mayor
NFGEQ	Necesariamente Mayor o Igual
NFLT	Necesariamente Menor
NFLEQ	Necesariamente Menor o Igual
NMGT	Necesariamente Mucho Mayor
NMLT	Necesariamente Mucho Menor

Tabla 2. Comparadores de necesidad.

3 Enfoque UML-Difuso

Ponemos a disposición de los diseñadores un patrón de diseño que hemos definido “*Atributos Difusos*”, que permite incluir o tratar atributos imprecisos de los tipos descritos previamente en las clases de sus diseños.

Rumbaugh et. al. [7] define un atributo en UML como la descripción de una ranura con el nombre de un tipo especificado en una clase; cada objeto de la clase tiene un valor independiente para el atributo. Un atributo se representa mediante una cadena de texto que puede dividirse en varias propiedades su notación es:

«estereotipo» visibilidad nombre multiplicidad : tipo = valor-inicial {cadena de propiedades}.

El enfoque propuesto se basa en la definición de los estereotipos «fuzzyt1», «fuzzyt2» y «fuzzyt3» para cada uno de los atributos difusos del Tipo 1, Tipo 2 y Tipo 3 respectivamente. Los estereotipos mencionados corresponden a clases que incorporan los métodos y atributos necesarios para manejar las propiedades difusas de estos datos, definidos como:

- La clase estereotipo «fuzzyt1» tiene asociado una lista de etiquetas lingüísticas, de modo de posibilitar el manejo de los atributos difusos Tipo 1.
- La clase estereotipo «fuzzyt2» se compone de una lista de trapecios que tienen asociada una etiqueta lingüística. El tipo de dato y valores de los atributos a, b, c y d, deben asignarse según el dominio del atributo difuso Tipo 2 que se utilizará.
- La clase estereotipo «fuzzyt3» tiene asociada una matriz que almacena los grados de similitud (gs) que se le atribuyen a los pares de etiquetas lingüísticas en consideración. No siempre la relación de similitud es simétrica, por lo que se diferencié el orden de las etiquetas en la relación (posición 1 y posición 2).

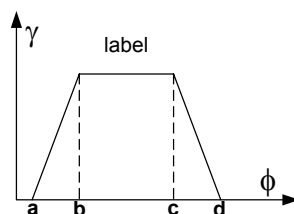


Figura 1. Representación de atributo Tipo 2.

Los atributos difusos Tipo 2 tienen asociado uno o más trapecios, como el que muestra la Figura 1. A cada valor entre a y d se le asocia un grado de pertenencia a label, además el valor de un atributo (sobre el referencial ordenado) puede considerar uno o más trapecios, obteniendo así el grado de pertenencia a cada una de ellos.

Para aplicar el estereotipo en un caso concreto, se debe hacer uso de la relación de especialización, y generar una clase del tipo de atributo requerido, y asignarle el tipo de dato correspondiente. Este proceso se realiza en la “vista de definición difusa”. En la vista estática del problema en cuestión se usará la clase como dominio del atributo difuso correspondiente, el cuál además será identificado con la etiqueta del estereotipo correspondiente. En la sección 4 se ejemplifica el uso del patrón “*Atributos Difusos*” y que utiliza los estereotipos definidos.

3.1 Vista Estática Patrón Atributos Difusos.

Para poder modelar datos imprecisos o difusos en UML, nuestra propuesta está basada en una especialización, de tal forma que, el atributo difuso es definido en la superclase y el dominio del tipo de atributo difuso (Tipo 1, Tipo 2 o Tipo3) es definidos en una subclase, obteniendo

así, en cada subclase un tipo de representación correspondiente a un atributo impreciso. Esta subclase puede ser tratada como una función de distribución o una función de similitud según corresponda a cada uno de los casos. Además, se establece una multiplicidad 1..*. La Figura 2 muestra la representación en notación definida anteriormente.

Comentarios

El atributo clásicos o crisp presente en fuzzyt1 y fuzzyt2 puede admitir valores precisos del tipo de dato especificado por el diseñador, además de los valores Unknown, Undefined y Null. El atributo etiqueta puede tomar como valor cualquiera de las etiquetas que están asociadas a la clase (que determina de ese modo el dominio de etiquetas posibles). Puede tomar el valor nulo cuando crisp tiene valor.

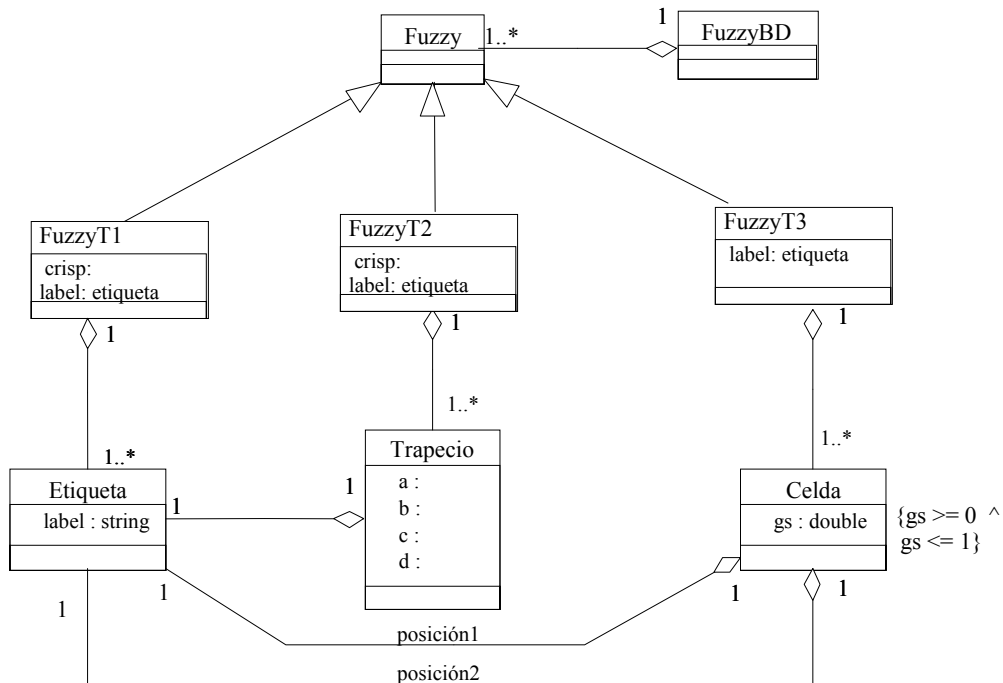


Figura 2. Patrón Atributos Difusos.

Consideramos que este tipo de representación de atributos difusos (Figura 2), en su notación muestra la semántica asociada a cada tipo de atributo, siendo un aporte a los diseñadores de bases de datos que contemplen en su modelado este tipo de expresiones. Además, el modelado en UML permite representar la estática correspondiente a los atributo, así como también, la dinámica que involucran los métodos.

3.2 Métodos Patrón Atributos Difusos.

Rumbaugh et. al. [7] definen un método como una implementación de una operación. Especifica el algoritmo o procedimiento que da lugar a los resultados de una operación.

Los métodos para cada clase del patrón de la Figura 4, se describen informalmente a continuación.

Clase *FuzzyBD*.

En este artículo nos enfocaremos sólo en la definición de métodos para consultar la base de datos difusa. Para ello usaremos las definiciones provistas por Galindo [3]. La definición de

los mecanismos de evaluación de las sentencias de la forma (Atributo Operador Valor) debe hacerse a priori y constituye un nivel de abstracción inferior en el diseño.

Restricción de Posibilidad (ClaseD, Atributo, Operador, Valor): Este método selecciona todos los objetos de la clase ClaseD que satisfacen (Atributo Operador Valor) donde Operador pertenece a {FEQ, FGT, FGEQ, FLT, FLEQ, MGT, MLT} y Valor pertenece al dominio de Atributo.

Restricción de Necesidad (ClaseD, Atributo, Operador, Valor): Este método selecciona todos los objetos de la clase ClaseD que satisfacen (Atributo Operador Valor) donde Operador pertenece a {NFEQ, NFGT, NFGEQ, NFLT, NFLEQ, NMGT, NMLT} y Valor pertenece al dominio de Atributo.

Consulta FuzzyT1 (ClaseT1, Atributo, Label): Este método selecciona todos los objetos de la clase difusa tipo 1 ClaseT1, tal que el valor del atributo del tipo Atributo esté asociado al rótulo Label.

Consulta FuzzyT2 (ClaseT2, Atributo, Label, gp): Este método selecciona todos los objetos de la clase difusa tipo 2 ClaseT2, tal que el valor del atributo del tipo Atributo esté asociado al rótulo Label con un grado de pertenencia mayor a gp.

Consulta FuzzyT3 (ClaseT3, Atributo, Label, gs): Este método selecciona todos los objetos de la clase difusa tipo 3 ClaseT3, tal que el atributo del tipo Atributo tenga un grado de similitud en relación al rótulo Label mayor a gs.

Clase Fuzzy.

Asociar Etiqueta: asocia una etiqueta de entre las asociadas a la clase, a una instancia específica.

Retornar Grado de Pertenencia (Label): retorna el grado de pertenencia de la instancia a la etiqueta lingüística Label (se especializa para cada subclase).

Clase: Fuzzyt1.

Retornar Crisp: indica cual es el valor exacto del atributo, cuando está asignado.

Asignar crisp: asigna un valor exacto a el atributo.

Retornar Grado de Pertenencia (Label): retorna 1 si la etiqueta lingüística está asociada a la instancia.

Clase: Fuzzyt2.

Retornar Crisp: indica cual es el valor exacto del atributo, cuando está asignado.

Asignar crisp: asigna un valor exacto a el atributo.

Retornar Grado de Pertenencia (Label): Evalúa el grado de pertenencia en base a la especificación dada en el trapecio correspondiente para el valor crisp de la instancia y la etiqueta especificada.

Clase Fuzzyt3.

Retornar Grado de Pertenencia (Label): En este caso, se evalúa la similitud entre la instancia y

la etiqueta Label, según lo especificado en la instanciación de la clase celda.

4 Ejemplo.

Supongamos una clase Persona en la cual se pueden definir atributos tales como, *edad* que puede tomar valores de “*infante*”, “*joven*” y “*adulto*”, por otro lado, se puede definir el atributo *color del pelo* que puede tomar valores de “*Rubio*”, “*Castaño*” y “*Pelirrojo*”.

En una base de datos clásica, cada instancia sólo puede tomar uno de estos tres valores, tanto para la *edad*, como para el *color del pelo*. Pero es usual que los requerimientos del usuario van más allá de éstas consideraciones, constantemente nos enfrentamos al hecho de que, una persona describe su edad como “*entre joven y adulto*” o que su color del pelo es “*entre rubio y castaño*”, consideramos que el modelado de datos en UML, no es capaz de representar este tipo de requerimientos. Es por ello, que nuestro trabajo se centra en la representación de este tipo de requerimientos usuales, más aún, se puede identificar que el dominio *edad* y *color del pelo* son distintos. Nos hemos propuesto el siguiente modelado en UML para este tipo de atributo que contiene la clase Persona, aplicando el *Patrón de Atributos Difuso* definidos en el párrafo anterior.

4.1 Ejemplo para atributos difusos tipo 2.

Para el caso del atributo *edad*, se propone definirlo como *atributo difuso Tipo 2*, considerando cada uno de sus valores como etiquetas lingüísticas dadas por: “*Infante*”, “*Joven*” y “*Adulto*”, definidas sobre un dominio ordenado entre 0 y 100 y una función de distribución (trapezoides) asociada a cada una de las etiquetas. Los valores de los atributos a, b, c y d correspondientes a cada uno de los trapezoides de un conjunto difuso son definidos como: *infante* = {0/1, 1/7, 1/10, 0/15}, *joven* = {0/10, 1/15, 1/25, 0/30} y *adulto* = {0/25, 1/30, 1/65, 1/100}, para cada una de las etiquetas se definen en la instanciación de la vista de definición difusa.

En la vista estática del diseño en cuestión, se define el atributo difuso *edad* dentro de la clase *Persona*, como se muestra en la Figura 3.

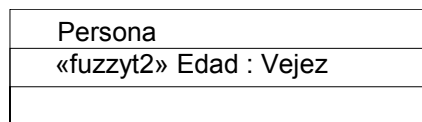


Figura 3. Vista estática con inclusión de un atributo difuso.

En el apartado 3.1 se definió que cada clase, para el caso que define atributos difusos, debe formar una generalización correspondiente a las etiquetas. La Figura 4, muestra la representación de la *edad* como atributo Tipo 2 definiendo el valor que toma cada uno de los conjuntos de las etiquetas definidas.

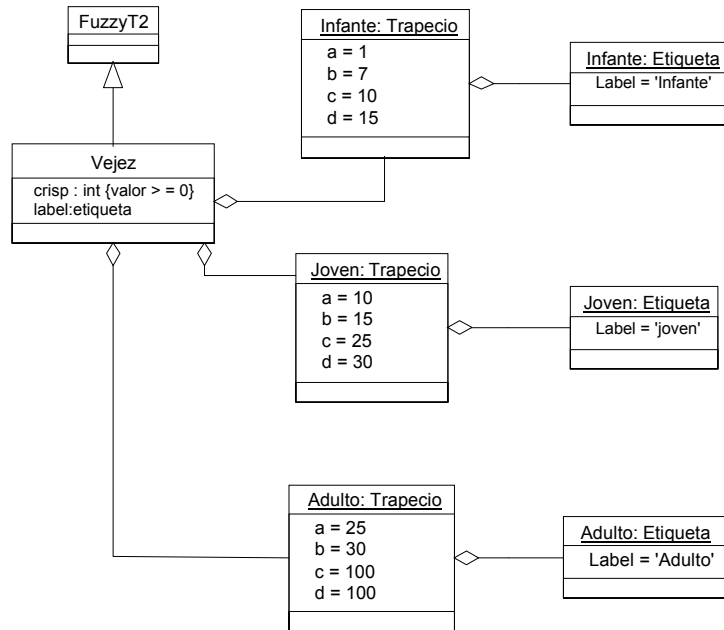


Figura 4. Vista de definición difusa para el caso propuesto.

La Figura 4, muestra una extensión del Tipo de atributo con referencial ordenado que puede estar definido en un modelo de datos en UML, además las etiquetas definidas para un atributo Tipo 2, pueden ser tantas como el usuario requiera, y los valores que estas tomen, pueden ser especificadas claramente en el modelo propuesto.

4.2 Ejemplo para atributo difuso tipo 3.

Para el caso del atributo *Color del pelo*, se propone definirlo como *atributo difuso Tipo 3*, para el cual se definen grados de similitud o semejanza entre cada una de las etiquetas “Rubio”, “Castaño” y “Pelirrojo”. Se ha establecido los grados de similaridad entre estas etiquetas según se muestra en la tabla 1.

Color de pelo	Rubio	Castaño	Pelirrojo
Rubio	1	0.1	0.5
Castaño	0.1	1	0.7
Pelirrojo	0.5	0.7	1

Tabla 1. Grados de similaridad entre color de pelo Rubio, Castaño y Pelirrojo.

En la vista estática del diseño en cuestión, se define el atributo difuso *color del pelo* dentro de la clase *Persona*, como lo muestra en la Figura 5.

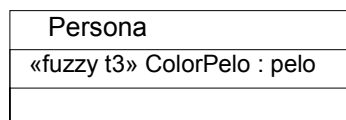


Figura 5. Vista estática con uso del estereotipo fuzzy tipo 3 para el atributo color de pelo.

El mapeo de la Tabla 1 al diseño en UML, considerando la notación que se ha definido para un atributo Tipo 3, se realiza según lo mostrado en la vista de definición difusa de la Figura 6.

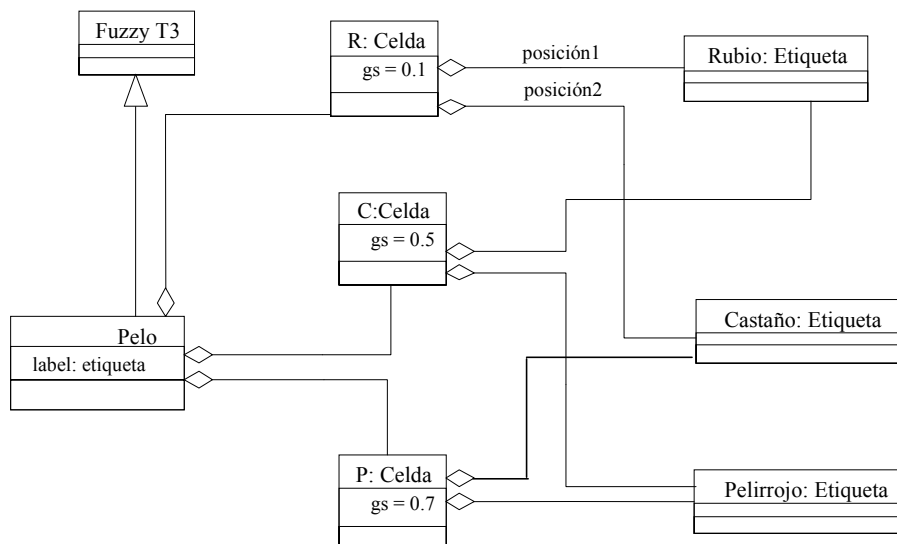


Figura 6. Vista de Definición Difusa para atributos difusos tipo 3.

La Figura 6, muestra una extensión del Tipo de atributo con referencial no ordenado que puede estar definido en un modelo de datos en UML, además los valores de similitud o semejanza definidos para un atributo Tipo 3, pueden ser tantas como el usuario requiera, y los valores que estas tomen, pueden ser especificadas claramente en el modelo propuesto.

4.3 Ejemplo de consultas difusas.

Consideremos que la base de datos difusa de personas posee la siguiente información.

Persona	Edad	Color de Pelo
P1	12	Pelirrojo
P2	19	Castaño
P3	28	Castaño
P4	35	Rubio

Se requiere a un joven pelirrojo. Considerando la especificación de las etiquetas lingüísticas, grados de pertenencia y similitud dados en los ejemplos previos, se define lo que sigue.

Consulta 1: Jóvenes con grado de pertenencia 0.8 de pelo pelirrojo con grado de similitud 0.8.

Utilizando los métodos especificados, dividimos la consulta en dos partes:

Parte 1: Consulta FuzzyT2(Persona, Edad, "Joven", 0.2)

Persona	Edad	Color de Pelo	Gp c/r Joven
P1	14	Pelirrojo	0.8
P2	19	Castaño	1
P3	28	Castaño	0.4
P4	35	Rubio	0

Respuesta Parte 1:

Persona	Edad	Color de Pelo	Gp c/r Joven
P1	14	Pelirrojo	0.8
P2	19	Castaño	1
P3	28	Castaño	0.4

Parte 2: Consulta FuzzyT3(Persona, Color de Pelo, "pelirrojo", 0.8)

Persona	Edad	Color de Pelo	Gs c/r pelirrojo
P1	14	Pelirrojo	1
P2	19	Castaño	0.7
P3	28	Castaño	0.7
P4	35	Rubio	0.5

Respuesta Parte 2:

Persona	Edad	Color de Pelo	Gs c/r pelirrojo
P1	14	Pelirrojo	1

La respuesta a la consulta 1 está dada por la intersección de Respuesta parte 1 y Respuesta Parte 2 :

Persona	Edad	Color de Pelo
P1	14	Pelirrojo

5 Conclusiones y trabajos futuros

La Teoría de conjuntos difusos permite acercar el funcionamiento de los sistemas de información al modo de trabajo de los seres humanos, pues las personas manejamos con gran frecuencia conceptos difusos (como “alto”, “bueno”, “malo”, “aproximadamente 8”...) que incluyen cierta imprecisión y que los sistemas informáticos tradicionales no entienden y, por tanto, no pueden utilizar.

Las bases de datos difusas han sido también ampliamente estudiadas con el objetivo de permitir el almacenamiento de datos imprecisos o difusos y la consulta de forma imprecisa de los datos existentes. Sin embargo, tradicionalmente la aplicación de la lógica difusa a las bases de datos ha prestado escasa atención al problema del modelado conceptual. La extensión del modelo UML para el tratamiento de datos difusos, ha sido estudiado en pocas publicaciones, pero en ninguna se referencia la posibilidad de modelar datos imprecisos en forma genérica utilizando etiquetas lingüísticas, referencias ordenado o de semejanza utilizando las herramientas que ofrece la teoría de conjuntos difusos.

En [9] los autores han propuesto un modelado de atributos imprecisos utilizando la notación EER, la gran diferencia con nuestra propuesta en UML, es que la semántica está presente en el modelado de estereotipos y representada en patrones de *Atributos Difusos*, no en el diccionario de datos como proponen los autores de [9], siendo UML más flexible y dando una gran expresividad al modelo de datos imprecisos con el tratamiento de la teoría de conjuntos difusos.

En este artículo se ha realizado una propuesta para expresar atributos que sus dominios sean imprecisos y la forma de representar estos en un modelo conceptual como UML. Las representaciones expuestas tienen un novedoso significado y aportan gran expresividad al modelo conceptual, incluso fácil de entender por usuarios no técnicos, algo que es fundamental en el modelado conceptual.

Además, se han definido las bases conceptuales que permiten especificar en UML bases de datos difusas que pueden implementarse utilizando la propuesta de Galindo [3] en lo referente al almacenamiento de datos difusos y su manipulación.

Como trabajo futuro se pretende depurar los estereotipos y patrón propuesto además de

seguir aplicando la teoría de conjuntos difusos al resto de las notaciones en UML, como son; multiplicidad, herencia etc., permitiendo así un modelado en UML aún más flexible. Por otra parte, se planea desarrollar un ambiente integrado que permita el diseño e implementación de bases de datos difusas objeto - relacionales.

6 Referencias

- [1] Geneste L., Ruet M., Fuzzy Case Based Configuration (2002): 15th European Conference on Artificial Intelligence, ECAI'2002, Workshop on Configuration, Lyon, France.
- [2] Geneste L., Ruet M. (2001): Experience based configuration, 17th International Conference on Artificial Intelligence, IJCAI'01, Workshop on Configuration, Seattle, Washington, USA, 4-10 august 2001.
- [3] Galindo J. (1999): Tratamiento de la Imprecisión en Bases de Datos Relacionales: Extensión del Modelo y Adaptación de los SGBD Actuales. Ph. Doctoral Thesis, University of Granada (Spain). (www.lcc.uma.es).
- [4] Larman C. (1999), "UML y Patrones", Prentice Hall.
- [5] Ma Z. M., Zhang W. J., Ma W. Y., Chen Q. (2001) Conceptual Design of Fuzzy Object-Oriented Databases Using Extended Entity-Relationship Model. International Journal of Intelligent System. Vol 16. pág 697-711, 2001
- [6] Marín N., Pons O., Vila M.A. (2000): Fuzzy Types: A New Concept of Type for Managing Vague Structures. International Journal of Intelligent Systems, 15, pp. 1061-1085.
- [7] Rumbaugh J., Jacobson J., Booch G., (1999): "The Unified Modeling Language Reference Manual", Addison Wesley.
- [8] Urrutia A., Piattini M. (2001): Transformation of imprecise data to linguistic labels for model E/R. Conferencia SCI/2001. 7th International Conference on Information System Analysis and Synthesis (ISAS2001), Pág. 351, 355. Orlando, USA.
- [9] Urrutia A., Galindo J. (2001): "Notación para datos con imprecisión en un modelo conceptual EER difuso", UC-Maule Revista Académica de la Universidad Católica del Maule, diciembre N° 27, pág. 39-48.
- [10] Zadeh L. A. (1965): Fuzzy Sets. Information and Control, 8, pp. 338-353.