

## PHƯƠNG PHÁP PHÁT HIỆN CÁC LỚP BỘ PHẬN TRONG LỚP ĐỐI TƯỢNG MỜ

ĐOÀN VĂN BAN<sup>1</sup>, HỒ CẨM HÀ<sup>2</sup>, VŨ ĐỨC QUẢNG<sup>3</sup>

<sup>1</sup>*Viện Công nghệ Thông tin, Viện Khoa học và Công nghệ Việt Nam*

<sup>2</sup>*Trường Đại học Sư phạm Hà nội*

<sup>3</sup>*Trường Đại học Quảng Nam*

**Abstract.** Bài báo tập trung nghiên cứu các loại quan hệ giữa các thuộc tính và giữa thuộc tính với phương thức trong một lớp đối tượng mờ. Mức độ gắn kết giữa các thuộc tính trong cơ sở dữ liệu mờ được xác định dựa trên khái niệm về phụ thuộc mạnh, yếu. Từ đó, chúng tôi đi phân tích lớp đối tượng mờ để phát hiện ra các lớp bộ phận tồn tại trong lớp làm cơ sở cho việc chuẩn hoá lớp nhằm hạn chế sự dư thừa dữ liệu.

**Tóm tắt.** In this paper, we focus on reseaching the types of relationship between attributes, between the attributes and methods in a fuzzy object class. The degree of coherence between the attributes is determined by rely on the concepts of strong and weak dependent relation. From there, we analyse fuzzy object classes to detect the component classes in them as the basis for normalizing the class to reduce data redundancy.

### 1. GIỚI THIỆU

Một trong những kết quả nghiên cứu quan trọng trong các hệ thống cơ sở dữ liệu là giảm đến mức có thể dư thừa dữ liệu. Vấn đề dư thừa dữ liệu đã được nghiên cứu nhiều trong các hệ thống cơ sở dữ liệu quan hệ/hướng đối tượng. Tuy nhiên, nó chưa được nghiên cứu nhiều trong hệ thống cơ sở dữ liệu hướng đối tượng mờ. Sự dư thừa dữ liệu rất có hại đến tính toàn vẹn dữ liệu và không đảm bảo tính nhất quán thông tin. Trên cơ sở mô hình hướng đối tượng mờ của Zongmin Ma [11], chúng tôi trình bày phương pháp phân tích để phát hiện ra các lớp bộ phận chứa trong các lớp của cơ sở dữ liệu hướng đối tượng mờ làm cơ sở cho việc chuẩn hoá lớp nhằm hạn chế sự dư thừa dữ liệu. Phương pháp của chúng tôi dựa vào phụ thuộc mờ giữa các thuộc tính, phụ thuộc mờ giữa thuộc tính và phương thức trong một lớp (mờ). Bằng việc sử dụng các phụ thuộc mờ đó các lớp bộ phận (mờ hoặc rõ) chứa trong lớp đối tượng mờ sẽ được phát hiện. Bài báo được tổ chức như sau: mục 2 trình bày cơ sở lý thuyết về tập mờ, các quan hệ ngữ nghĩa và ước lượng ngữ nghĩa giữa hai tập mờ; mục 3 nêu các định nghĩa về các phụ thuộc mờ, các luật suy dẫn, trình bày qui trình phát hiện các lớp bộ phận; cuối cùng là phần kết luận.

## 2. CÁC KHÁI NIỆM CƠ BẢN

### 2.1. Tập mờ và phân bố khả năng

Theo Zadeh [10], dữ liệu mờ được mô tả như một tập mờ. Cho  $U$  là một vũ trụ, một tập mờ  $F$  trên  $U$  xác định bởi hàm thuộc  $\mu_F : U \rightarrow [0, 1]$ , gán mỗi phần tử  $u$  thuộc  $U$  một giá trị để chỉ độ thuộc của  $u$  vào tập mờ  $F$ . Tập mờ  $F$  được biểu diễn dưới dạng

$$F = \{\mu_F(u_1)/u_1, \mu_F(u_2)/u_2, \mu_F(u_3)/u_3, \dots, \mu_F(u_n)/u_n\} \quad \text{hay} \quad F = \int_{u \in U} \frac{\mu_F(u)}{u}.$$

Khi  $\mu_F(u)$  được xem như độ đo khả năng mà một biến  $X$  nhận giá trị  $u$ , một giá trị mờ được biểu diễn bằng phân bố khả năng  $\pi_X$  như sau

$$\pi_X = \{\pi_X(u_1)/u_1, \pi_X(u_2)/u_2, \pi_X(u_3)/u_3, \dots, \pi_X(u_n)/u_n\},$$

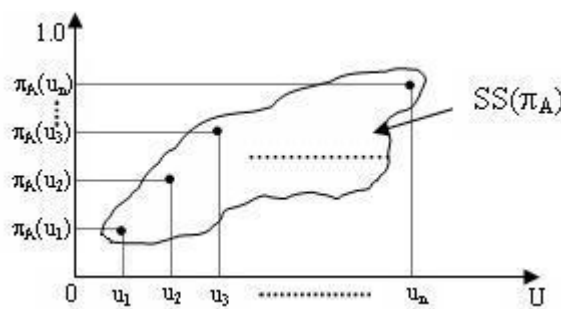
trong đó,  $\pi_X(u_i)$ ,  $u_i \in U$ , biểu thị khả năng  $X$  nhận giá trị  $u_i$ ,  $\pi_X, F$  lần lượt biểu diễn phân bố khả năng và tập mờ đối với một giá trị mờ. Rõ ràng,  $\pi_X = F$ .

### 2.2. Quan hệ ngữ nghĩa giữa hai dữ liệu mờ

Cho tập vũ trụ  $U = \{u_1, u_2, \dots, u_n\}$ ,  $A$  là biến mờ, giá trị mờ của  $A$  trên miền  $U$  theo phân bố khả năng là  $\pi_A = \{\pi_A(u_1)/u_1, \pi_A(u_2)/u_2, \pi_A(u_3)/u_3, \dots, \pi_A(u_n)/u_n\}$ ,  $u_i \in U, \pi_A(u_i) \in [0, 1]$ .

**Định nghĩa 2.1.** [11] Ngữ nghĩa của dữ liệu mờ  $\pi_A$  tương ứng với một miền trong không gian  $(U \times [0, 1])$ , trong đó tập vũ trụ tương ứng với trục  $X$ , khả năng tương ứng với trục  $Y$ .

Ngữ nghĩa của dữ liệu mờ  $\pi_A$  còn được gọi là không gian ngữ nghĩa, ký hiệu là  $SS(\pi_A)$  và biểu diễn như sau.



Quan hệ ngữ nghĩa giữa hai dữ liệu mờ được mô tả bởi quan hệ giữa hai không gian ngữ nghĩa của chúng. Cho hai dữ liệu mờ  $\pi_A$  và  $\pi_B$ ,  $SS(\pi_A)$  và  $SS(\pi_B)$  lần lượt là không gian ngữ nghĩa của chúng. Nếu  $SS(\pi_A) \supseteq SS(\pi_B)$ , ta nói  $\pi_A$  bao hàm ngữ nghĩa  $\pi_B$  hay  $\pi_B$  được bao hàm ngữ nghĩa bởi  $\pi_A$ . Nếu  $SS(\pi_A) \supseteq SS(\pi_B)$  và  $SS(\pi_A) \subseteq SS(\pi_B)$ , ta nói  $SS(\pi_A)$  và  $SS(\pi_B)$  là tương đương ngữ nghĩa với nhau.

**Định nghĩa 2.2.** [11] Độ bao hàm ngữ nghĩa.  $\pi_A$  và  $\pi_B$  là hai dữ liệu mờ, không gian ngữ nghĩa của chúng lần lượt là  $SS(\pi_A)$  và  $SS(\pi_B)$ . Ký hiệu  $SID(\pi_A, \pi_B)$  là mức độ mà  $\pi_A$  bao hàm ngữ nghĩa  $\pi_B$ , được biểu thị như sau

$$SID(\pi_A, \pi_B) = (SS(\pi_A) \cap SS(\pi_B)) / SS(\pi_B),$$

trong đó,  $(SS(\pi_A) \cap SS(\pi_B)) / SS(\pi_B)$  là giao hai không gian ngữ nghĩa của hai dữ liệu mờ  $\pi_A$  và  $\pi_B$  trên không gian ngữ nghĩa của dữ liệu mờ  $\pi_B$ ,  $SID(\pi_A, \pi_B)$  được xem là tỷ lệ phần trăm không gian ngữ nghĩa của  $\pi_B$  được bao hàm trong không gian ngữ nghĩa của  $\pi_A$ .

**Định nghĩa 2.3.** [11] (Độ tương đương ngữ nghĩa) Cho hai dữ liệu mờ  $\pi_A$  và  $\pi_B$ . Mức độ mà  $\pi_A$  và  $\pi_B$  tương đương ngữ nghĩa với nhau, ký hiệu là  $SE(\pi_A, \pi_B)$  và được biểu thị như sau

$$SE(\pi_A, \pi_B) = \min(SID(\pi_A, \pi_B), SID(\pi_B, \pi_A)).$$

### 2.3. Ước lượng độ đo ngữ nghĩa

**Định nghĩa 2.4.** [11] Cho  $U = \{u_1, u_2, \dots, u_n\}$  là tập vũ trụ.  $\pi_A$  và  $\pi_B$  là hai dữ liệu mờ trên  $U$  dựa trên phân bố khả năng.  $\pi_A(u_i), u_i \in U$  biểu thị khả năng mà  $A$  nhận giá trị  $u_i$ . Khi đó  $SID(\pi_A, \pi_B)$  được xác định như sau

$$SID(\pi_A, \pi_B) = \sum_{i=1}^n \min_{u_i \in U} (\pi_B(u_i), \pi_A(u_i)) / \sum_{i=1}^n \pi_B(u_i).$$

**Định nghĩa 2.5.** [11] Cho  $U = \{u_1, u_2, \dots, u_n\}$  là tập vũ trụ.  $\pi_A$  và  $\pi_B$  là hai dữ liệu mờ trên  $U$  dựa trên phân bố khả năng.  $\pi_A(u_i), u_i \in U$  biểu thị khả năng mà  $A$  nhận giá trị  $u_i$ . Res là một quan hệ tương đương trên miền  $U$ ,  $\alpha (0 \leq \alpha \leq 1)$  là một ngưỡng tương ứng với Res.  $SID(\pi_A, \pi_B)$  được xác định như sau

$$SID(\pi_A, \pi_B) = \sum_{i=1}^n \min_{u_i, u_j \in U, Res(u_i, u_j) \leq \alpha} (\pi_B(u_i), \pi_A(u_i)) / \sum_{i=1}^n \pi_B(u_i),$$

trong đó, quan hệ tương tự Res (Resemblance relation) trên miền  $U$  là một ánh xạ  $U \times U \rightarrow [0, 1]$  thoả mãn các tính chất sau

- (1)  $\forall u \in U, Res(u, u) = 1$  (Tính phản xạ);
- (2)  $\forall u_i, u_j \in U, Res(u_i, u_j) = Res(u_j, u_i)$  (Tính đối xứng).

## 3. PHƯƠNG PHÁP XÁC ĐỊNH CÁC LỚP BỘ PHẬN TRONG LỚP ĐỐI TƯỢNG MỜ

Dựa trên mô hình cơ sở dữ liệu hướng đối tượng mờ được đưa ra bởi Zongmin Ma [11], một lớp mờ  $C$  bao gồm tập các thuộc tính và các phương thức

$$C = (\{a_1, a_2, \dots, a_k\}, \{M_1, M_2, \dots, M_m\}, \{w_1, w_2, \dots, w_k\}),$$

trong đó,  $a_i$  là một thuộc tính mờ (hoặc rõ),  $M_j$  là một phương thức,  $w_i \in (0, 1]$  là trọng số của thuộc tính  $a_i$  tương ứng. Một thuộc tính bao gồm tên  $n$ , kiểu  $t$  và trọng số  $w_i$

$$a_i = \langle n, t, w_i \rangle$$

Các kiểu thuộc tính được phân thành các loại sau:

- (1) Kiểu đơn giản (simple type): là các kiểu cơ sở (integer, string, real, boolean), kiểu lớp rõ (hoặc mờ).
- (2) Kiểu sưu tập (collection type): là một danh sách hay một tập các thuộc tính có cùng kiểu (kiểu đơn giản)
- (3) Kiểu nhóm (group type): Thuộc tính chứa một nhóm các giá trị khác kiểu.

Phương thức định nghĩa trong lớp được mô tả như sau:

$$M_j(N, I, R) \Rightarrow (u, v, g)$$

, trong đó:

$N$ : tên phương thức;

$I$ : tập các tham số đầu vào  $\{\langle \text{tên}, \text{kiểu} \rangle\}$ ;

$R$ : tập các thuộc tính mà giá trị của nó được đọc bởi phương thức;

$u$ : tập các tham số đầu ra bao gồm kiểu giá trị trả về  $\{\langle \text{tên}, \text{kiểu} \rangle\}$ ;

$v$ : tập các thuộc tính mà giá trị của nó bị thay đổi bởi phương thức;

$g$ : tập các thông báo được đưa ra bởi phương thức có dạng  $\{[o, msg, p]\}$ ,  $o$  là nơi nhận thông báo,  $msg$  là thông báo và  $p$  là tập các tham số có trong thông báo  $\{\langle n, t \rangle\}$ .

**Ví dụ 1.** Cho một lớp Kiemke được minh họa như hình dưới đây. Lớp này thu thập thông tin về: cửa hàng cung cấp (cuahang), nhân viên giao dịch của cửa hàng (nhanvien), kho chứa hàng, bảng kê số lượng hàng. Thông tin về bảng kê số lượng hàng bao gồm: mã số hàng (masohang), tình trạng hàng (tinhtrang), chi phí trên một đơn vị hàng (chiphi), giá bán lẻ (giabanle), số lượng (soluong), ngày nhập hàng (ngaynhap). Mô hình cơ sở dữ liệu mà ta phân tích giả sử được đóng gói hoàn toàn, tất cả các thuộc tính đều không được truy xuất bởi các đối tượng khác.

Class Kiemke with degree of 1.0

ATTRIBUTE

masohang: type of string;

tinhtrang: fuzzy domain{tot, vua, xau} type of string;

chiphi: fuzzy type of real;

giabanle: fuzzy type of real;

soluong: type of integer;

ngaynhap: type of integer;

cuuhang: type of Ctycungcap;

nhanvien: type of set(Nguoi) //nhân viên giao dịch;

dckhohang: type of Diachi //địa chỉ kho hàng;

masokho: type of string;

ngqly: type of Ngquanly;

WEIGH

...

## METHODS

nhaphang(soluong: type of integer);  
 suagiaban(gia-moi: fuzzy type of real);  
 suanhanvien(nhanvien-gdich: Nguoi);  
 suadckho(diachi-kho: Diachi);  
 baocaokke( );  
 thongkenhap( );  
 thongkekho( );

...

pD: domain[0,1] of real;

END

Sau đây là một thể hiện của lớp Kiemke, Ctycongcap, Diachi, Nguoi, Ngquanly. Để đơn giản ta giới hạn bảng dữ liệu chỉ gồm những thuộc tính.

Bảng 1. Thể hiện của Lớp Kiemke

Maso hang	Tinhtrang	Chiph	Giabanle	So luong	Ngaynhap	Maso kho	Dckho Hang	Ngqly	Cua hang	Nhanvien
1#	{0.9/tot,0.8/vua,0.4/xau}	{0.7/100,0.5/80}	{0.8/10000,0.6/12000}	9	12/06/2010	1#	id1	iD1	Id1	ID1,ID2
1#	{0.8/tot,0.2/vua,0.5/xau}	{0.9/100,0.2/80}	{0.7/10000,0.5/12000}	12	14/06/2010	2#	id2	iD2	Id1	ID1,ID2
1#	{9,0/tot,0.7/vua,0.5/xau}	{0.8/100,0.3/80}	{0.6/10000,0.7/12000}	9	12/06/2010	2#	id2	iD2	Id1	ID1,ID2
1#	{0.8/tot,0.2/vua,0.4/xau}	{0.8/100,0.2/80}	{0.8/10000,0.4/12000}	12	14/06/2010	1#	id1	iD1	Id1	ID1,ID2

Bảng 2. Thể hiện của lớp Nguoi và lớp Diachi

IDNguoi	Hoten	Ngaysinh
ID1	Nam	17/02/1980
ID2	Quang	20/08/1979

idDiachi	Duong	So	Tinh	Quocgia
id1	HungVuong	102	HaNoi	VietNam
id2	Le Loi	100	QuangNam	VietNam

Bảng 3. Thể hiện của lớp Ctycongcap và lớp Nguoiqly

IdCuahang	Tencongty	Sodt	Taikhon
Id1	Nam Hải	043677535	01237094
Id2	Tiến Minh	083766544	01237097

iDngqly	Hoten	Ngaysinh	Sdt
iD1	Tam	043677535	01237094
iD2	Ky	083766544	01237097

Tương tự như mô hình hướng đối tượng, mô hình cơ sở dữ liệu hướng đối tượng mờ là mô hình dữ liệu, trong đó các thuộc tính dữ liệu là mờ (hoặc rõ) và các phương thức thao tác trên các thuộc tính đó đều được đóng gói trong các cấu trúc gọi là đối tượng (mờ).

Tập dữ liệu được tổ chức theo cấu trúc của mô hình dữ liệu hướng đối tượng mờ, chúng có thể có quan hệ mờ với nhau được gọi là CSDL hướng đối tượng mờ (Fuzzy object-oriented

database, viết tắt FOODB). Theo hướng tiếp cận tương tự như trong cơ sở quan hệ mờ/rõ về sự gắn kết giữa các thuộc tính, ta đưa ra định nghĩa về phụ thuộc thuộc tính mờ trong một lớp đối tượng (fuzzy attribute dependency, viết tắt là fa-phụ thuộc).

### 3.1. FA-phụ thuộc

**Định nghĩa 3.1.** (Fa-phụ thuộc) Cho lớp mờ  $C$  với tập các thuộc tính  $U$  (ký hiệu  $C(U)$ );  $X, Y \in U, Z = U \setminus XY$ ,  $Y$  là fa-phụ thuộc vào  $X$  nếu mỗi một giá trị tương tự nhau trên  $X$  có một giá trị tương ứng trên  $Y$  mà không liên quan gì đến  $Z$  trong tất cả các đối tượng của lớp  $C$ , ký hiệu:  $X \xrightarrow{fa} Y$ .

Định nghĩa fa-phụ thuộc tương đương với định nghĩa phụ thuộc hàm đa trị mờ trong các cơ sở dữ liệu quan hệ mờ trong [1, 5, 8, 11] với một số mở rộng:

(1) Nếu cả  $X$  và  $Y$  là có kiểu đơn giản (nguyên tố) thì fa-phụ thuộc là phụ thuộc đa trị mờ, tức là  $\forall o_1, o_2$  là các đối tượng của lớp  $C$  thì  $\exists o_3$  của lớp  $C$  mà  $SE(o_3.X, o_1.X) \geq SE(o_1.X, o_2.X)$  và  $SE(o_3.Y, o_1.Y) \geq SE(o_1.X, o_2.X)$ ,  $SE(o_3.Z, o_2.Z) \geq SE(o_1.X, o_2.X)$

(2) Nếu  $X(Y)$  là một đối tượng (không nguyên tố) thì fa-phụ thuộc tương đương với phụ thuộc đa trị mờ khi đối tượng được thay bằng định danh của nó. Độ tương đương ngữ nghĩa trên  $X(Y)$  được tính trên các thuộc tính thông qua định danh của đối tượng  $X(Y$  tương ứng).

(3)  $X(Y)$  là một nhóm hay một tập hợp các thuộc tính  $[x_1, x_2, \dots, x_n]$  (các thuộc tính  $[y_1, y_2, \dots, y_n]$  tương ứng của  $Y$ ). Nếu  $\forall x_i(y_j)$  có kiểu đơn giản thì fa-phụ thuộc là phụ thuộc đa trị mờ. Nếu  $x_i(y_j)$  là một đối tượng, fa-phụ thuộc là tương đương với phụ thuộc đa trị mờ sau khi mỗi đối tượng  $x_i(y_j$  tương ứng) được thay thế bằng chính định danh của nó.

**Định lý 3.1.** *Phụ thuộc đa trị trong cơ sở quan hệ thoả mãn định nghĩa Fa-phụ thuộc với một số mở rộng (như trên).*

*Chứng minh:* Cho quan hệ  $r(U)$  thoả mãn phụ thuộc hàm đa trị:  $X \rightarrow Y$ ,  $X, Y \subseteq U, Z = U \setminus XY$ .  $\forall t_1, t_2, t_1[X] = t_2[X]$ ,  $\exists t_3$  và  $t_3[X] = t_1[X], t_3[Y] = t_1[Y]$ ,  $t_3[Z] = t_2[Z]$ . Giả sử  $U$  là tập các thuộc tính của lớp mờ  $C$  thì  $\forall o_1, o_2$  là các đối tượng của lớp  $C$  và tồn tại đối tượng  $o_3 \in C$ , ta có  $SE(o_3.X, o_1.X) = SE(o_3.Y, o_1.Y) = 1$ ,  $SE(o_3.Z, o_2.Z) = SE(o_1.X, o_2.X) = 1$ .

Suy ra:  $SE(o_3.X, o_1.X) \geq SE(o_1.X, o_2.X) \geq 1$ ,  $SE(o_3.Y, o_1.Y) \geq SE(o_1.X, o_2.X) \geq 1$ ,  $SE(o_3.Z, o_2.Z) \geq SE(o_1.X, o_2.X) \geq 1$ . Theo định nghĩa Fa-phụ thuộc ta có  $X \xrightarrow{fa} Y$  ■

**Ví dụ 2.** Một số fa-phụ thuộc trong lớp Kiemke.

{masohang, tinhtrang, cuahang, soluong, ngayban}  $\xrightarrow{fa}$  {chiphil, giabanle}

{cuahang}  $\xrightarrow{fa}$  {nhanvien}

{masohang, tinhtrang, cuahang, soluong, ngaynhap}  $\xrightarrow{fa}$  {masokho, dckhohang}

{masokho, dckhohang}  $\xrightarrow{fa}$  {ngqly}

Phụ thuộc {masohang, tinhtrang, cuahang, soluong, ngaynhap}  $\xrightarrow{fa}$  {chiphil, giabanle} thoả mãn định nghĩa Fa-phụ thuộc. Đặt  $o_1, o_2, o_3, o_4$  lần lượt là 4 đối tượng thuộc lớp Kiemke,  $X = \{\text{masohang, tinhtrang, cuahang, soluong, ngaynhap}\}$ ,  $Y = \{\text{chiphil, giabanle}\}$ ,  $Z = \{\text{masokho, dckhohang, ngqly, nhanvien}\}$ . Ta có:

$$SE(o_3.X, o_1.X) = \min\{SID(o_3.X, o_1.X), SID(o_1.X, o_3.X)\} \approx \min(0.95, 0.95) = 0.95$$

$$SE(o_1.X, o_2.X) = \min\{SID(o_1.X, o_2.X), SID(o_2.X, o_1.X)\} \approx \min(0.66, 0.93) = 0.66$$

$$SE(o_3.Y, o_1.Y) = \min\{SID(o_3.Y, o_1.Y), SID(o_1.Y, o_3.Y)\} \approx \min(0.83, 0.85) = 0.83$$

$$SE(o_3.Z, o_2.Z) = \min\{SID(o_3.Z, o_2.Z), SID(o_2.Z, o_3.Z)\} \approx \min(0.83, 0.85) = 1.0$$

Do đó,  $SE(o_3.X, o_1.X) \geq SE(o_1.X, o_2.X)$ ,  $SE(o_3.Y, o_1.Y) \geq SE(o_1.X, o_2.X)$  và  $SE(o_3.Z, o_2.Z) \geq SE(o_1.X, o_2.X)$ . Theo định nghĩa Fa-phụ thuộc ta có  $X \xrightarrow{fa} Y$

Việc kiểm tra các Fa-phụ thuộc còn lại thực hiện tương tự.

Giống như chuẩn hoá quan hệ (mờ), ta có thể tính toán tập đóng bất cầu các fa-phụ thuộc để mỗi tập đóng bất cầu tạo thành một lớp bộ phận mờ. Tuy nhiên, việc tính toán các tập đóng bất cầu cần được hạn chế. Không phải tất cả các fa-phụ thuộc đều biểu thị cùng mức độ gắn kết giữa các thuộc tính của một lớp bộ phận mờ. Qua phân tích ta nhận thấy dư thừa dữ liệu có nguyên nhân là do bỏ qua mức độ gắn kết. Vì vậy, ta phải xem xét mức độ gắn kết trong lúc tính toán tập đóng bất cầu và chỉ sử dụng các fa-phụ thuộc mà biểu thị gắn kết mạnh của các thuộc tính để tạo ra một lớp bộ phận.

Cho  $U$  là tập các thuộc tính của lớp  $C$  và  $X, Y, Z, W \in U$ , dựa vào định lý trên và các kết quả của [3, 6, 8, 11], ta có các luật suy dẫn fa-phụ thuộc sau.

**Luật 3.1.** Phản xạ. Nếu  $Y \subseteq X$  thì  $X \xrightarrow{fa} Y$ . Luật 1.1 dễ dàng được chứng minh dựa trên định nghĩa fa-phụ thuộc, các fa-phụ thuộc dạng này là yếu vì  $Y$  là tập con của  $X$  thì phụ thuộc này luôn tồn tại. Phụ thuộc này không phát hiện ra bất kỳ một phụ thuộc mới nào nên cần được loại bỏ.

**Luật 3.2.** Bù. Nếu  $X \xrightarrow{fa} Y$  thì  $X \xrightarrow{fa} Z - XY$ . Các fa-phụ thuộc này tồn tại tầm thường bởi định nghĩa fa-phụ thuộc nên cần được loại bỏ.

**Luật 3.3.** Gia tăng. Nếu  $X \xrightarrow{fa} Y$  và  $Z \in W$  thì  $XW \xrightarrow{fa} YZ$  là yếu. Luật này có thể được suy dẫn từ luật 1.1. Vì  $Z$  là tập con của  $W$ , sự xuất hiện của  $Z$  bên vế phải của luật là dư thừa và có thể được loại bỏ.

**Luật 3.4.** Bất cầu. Nếu  $X \xrightarrow{fa} Y$  và  $Y \xrightarrow{fa} Z$  thì  $X \xrightarrow{fa} Z - Y$ . Phụ thuộc này biểu thị gắn kết yếu, vì vậy nên loại bỏ.

**Luật 3.5.** Kết hợp. Nếu  $X \xrightarrow{fa} Y$  và  $Y \xrightarrow{fa} Z$  thì  $X \xrightarrow{fa} YZ$ . Việc hợp 2 fa-phụ thuộc này là dư thừa và không phơi bày ra bất kỳ phụ thuộc mới nào, các Fa-phụ thuộc này cần được loại bỏ.

**Luật 3.6.** Bắc cầu giả. Nếu  $X \xrightarrow{fa} Y$  và  $YW \xrightarrow{fa} Z$  thì  $XW \xrightarrow{fa} Z - YW$ . Luật này có thể được suy dẫn từ luật 1.4, fa-phụ thuộc này biểu thị gắn kết yếu nên được loại bỏ.

**Luật 3.7.** Giao. Nếu  $X \xrightarrow{fa} Y$  và  $X \xrightarrow{fa} Z$  thì  $X \xrightarrow{fa} Y \cap Z$ . Luật này có thể suy dẫn từ định nghĩa, bởi vì luật không biểu thị một gắn kết mới nào nên nó cần được loại bỏ.

**Luật 3.8.** Dẫn. Nếu  $X \xrightarrow{fa} Y$  và  $X \xrightarrow{fa} Z$  thì  $X \xrightarrow{fa} Y - Z$  và  $X \xrightarrow{fa} Z - Y$ . Hai luật được suy dẫn từ luật 1.7, hai luật này không biểu thị sự gắn kết mới nên nó cần được loại bỏ.

### 3.2. FM-phụ thuộc

Mối quan hệ giữa phương thức đối tượng với các thuộc tính được xác định thông qua việc phương thức sử dụng các thuộc tính đó để đọc hay sửa đổi. Sự phụ thuộc của phương

thức vào các thuộc tính như thế được gọi là phụ thuộc phương thức mờ (fuzzy method dependency – viết tắt  $fm$ -phụ thuộc).

**Định nghĩa 3.2.** ( $fm$ -phụ thuộc) Cho  $M_j$  là một phương thức của một lớp  $C$ , và  $X = R \cup v \neq \emptyset$  là tập các thuộc tính (đọc, sửa đổi) được sử dụng bởi  $M_j$ . Quan hệ giữa  $M_j$  và  $X$  được gọi là  $fm$ -phụ thuộc, ký hiệu  $X \xrightarrow{fm} M_j$

**Ví dụ 3.** Một số  $fm$ -phụ thuộc trong lớp Kiemke

$$\begin{aligned} & \{\text{masohang, tinhtrang, ngaynhap, cuuhang, giabanle, chiphi}\} \xrightarrow{fm} \{\text{nhaphang}(\dots)\} \\ & \{\text{chiphi, giabanle}\} \{\text{suagiaban}(\dots)\}; \{\text{cuahang, nhanvien}\} \xrightarrow{fm} \{\text{suacuuhang}(\dots)\} \\ & \{\text{nhanvien}\} \xrightarrow{fm} \{\text{suanhanvien}(\dots)\}; \\ & \{\text{dckhohang}\} \xrightarrow{fm} \{\text{suadckho}(\dots)\} \\ & \{\text{masohang, soluong, ngaynhap, masokho, dckhohang}\} \xrightarrow{fm} \{\text{thongkenhap}(\ )\} \\ & \{\text{masohang, tinhtrang, cuahang, soluong, masokho, dckhohang}\} \xrightarrow{fm} \{\text{thongkekho}(\ )\} \end{aligned}$$

Các luật suy dẫn của  $fm$ -phụ thuộc

**Luật 3.9.** Nếu  $X \xrightarrow{fm} M_j$  và  $Y \xrightarrow{fm} M_j$  thì  $XY \xrightarrow{fm} M_j$

**Luật 3.10.** Nếu  $X \xrightarrow{fm} M_i$  và  $X \xrightarrow{fm} M_j$  thì  $X \xrightarrow{fm} M_i M_j$

**Luật 3.11.** Nếu  $X \xrightarrow{fm} M_j$  và  $Y \subseteq X$  thì  $Y \xrightarrow{fm} M_j$

**Luật 3.12.** Nếu  $M_j$  gọi đến  $M_i$  và  $X \xrightarrow{fm} M_i, Y \xrightarrow{fm} M_j$  thì  $XY \xrightarrow{fm} M_j$

$fm$ -phụ thuộc giúp loại bỏ các thuộc tính không được sử dụng trong bất kỳ phương thức nào. Lý do ở đây là mỗi thuộc tính của lớp phải được sử dụng bởi ít nhất một phương thức trong lớp và bởi vì việc đóng gói dữ liệu, các thuộc tính không được sử dụng sẽ không được sử dụng từ bên ngoài lớp.

**Luật 3.13.**  $U$  là tập các thuộc tính của lớp mờ  $C$ , một thuộc tính  $a \in U$ , nếu không tồn tại một phương thức  $M_j$  nào trong  $C$  mà  $\{a\} \xrightarrow{fm} M_j$  thì tất cả các  $fa$ -phụ thuộc của  $C$  có dạng  $X_i \xrightarrow{fa} Y_i$  phải được giảm thành  $X_i - \{a\} \xrightarrow{fa} Y_i - \{a\}$ .

### 3.3. Lớp bộ phận mờ

Sự tồn tại một số lớp bộ phận (mờ hoặc rõ) trong cùng một lớp có thể dẫn đến dư thừa dữ liệu. Giống như các cơ sở dữ liệu khác, dư thừa dữ liệu làm cho thông tin trong cơ sở dữ liệu có thể bị tổn thất, không nhất quán. Trong một lớp mờ có thể chứa nhiều hơn một lớp bộ phận, khi có một lớp bộ phận xuất hiện trong một lớp đối tượng, các lớp bộ phận khác đi cùng có thể được lặp lại. Chẳng hạn như trong lớp Kiemke dư thừa dữ liệu xuất hiện là do lớp bộ phận mờ Nhakho trong lớp Kiemke có thể giống nhau trong một số mặt hàng (lớp bộ phận Hang) được chứa trong một kho hàng, ... Vì vậy, việc nhận ra các lớp bộ phận trong các lớp là bước đầu tiên để phân tích dư thừa dữ liệu. Để mô tả một lớp bộ phận trong lớp đối tượng mờ, sau đây chúng tôi đưa ra định nghĩa biên lớp bộ phận.

**Định nghĩa 3.3.** Biên lớp bộ phận  $C_i$  trong một lớp mờ  $C(U)$  là một bộ bốn gồm tập các thuộc tính, các  $fa$ -phụ thuộc, các phương thức và các  $fm$ -phụ thuộc, ký hiệu:

$$C_i = (A_i, \psi, M_i, \pi_i)$$



trong đó,

$A_i \subseteq U$  : tập các thuộc tính mờ (hoặc rõ);

$\psi_i$  : tập các fa-phụ thuộc mạnh;

$S_i \xrightarrow{fa} Y_j$ , và  $S \subset A_i, Y_j \subset A_i$ ;

$M_i$ : tập các phương thức;

$\pi_i$ : tập các  $fm$ -phụ thuộc  $X \xrightarrow{fm} M_{i_k}, M_{i_k} \in M_i, X \subseteq A_i$ .

Theo Định nghĩa 3.3, tất cả các phụ thuộc có chung về trái ( $S$ ),  $S$  giống như là siêu khoa trong các cơ sở dữ liệu quan hệ và  $S$  được sử dụng như một tham chiếu đến lớp bộ phận.  $S$  được gọi là ký hiệu ngữ nghĩa của lớp bộ phận.

*Ví dụ 4.* Biên lớp bộ phận Nhakho trong lớp Kiemke

$A_i = \{\text{masokho, dckhohang, ngquanly}\}$ ,

$\psi_i = \{\text{masokho, dckhohang}\} \xrightarrow{fa} \{\text{ngquanly}\}$ ,

$\pi_i = \{\{\text{masokho, dckhohang}\} \xrightarrow{fm} \text{thongkekho(...), suadckho(...)}\}$ .

**Định nghĩa 3.4.** Ký hiệu ngữ nghĩa của một lớp bộ phận là tập các thuộc tính giống nhau ở về trái của tất cả các fa-phụ thuộc của một lớp bộ phận.

Việc xác định các ký hiệu ngữ nghĩa  $S$  của lớp bộ phận không chính xác có thể đưa đến kết quả biên lớp bộ phận quá hẹp hay quá rộng. Biên quá hẹp có thể loại bỏ các thuộc tính có nghĩa và dẫn đến kết quả các lớp bộ phận bị phân mảnh. Ngược lại, một biên quá rộng có thể bao hàm các thuộc tính không cần thiết và không có khả năng để phát hiện dư thừa dữ liệu tiềm ẩn. Vì vậy, ký hiệu ngữ nghĩa của lớp bộ phận mờ phải được kiểm tra trước khi chúng được sử dụng để xác định các lớp bộ phận.

Để kiểm tra và cấu trúc lại các fa-phụ thuộc, chúng tôi sử dụng 3 luật suy dẫn sau để khắc phục các lỗi trên. Luật thứ nhất là nói rộng ký hiệu ngữ nghĩa của lớp bộ phận để các lớp bộ phận bao hàm các thuộc tính cần thiết, 2 luật tiếp theo làm giảm của lớp bộ phận.

**Luật 3.14.** Gia tăng. Cho  $X, Y, W$  là 3 tập khác rỗng. Nếu  $X \xrightarrow{fa} Y, Y \xrightarrow{fa} W$  thì  $XY \xrightarrow{fa} W$ .

Luật này được suy ra từ định nghĩa fa-phụ thuộc. Nếu  $Y$  là fa-phụ thuộc vào  $X$  và  $W$  là fa-phụ thuộc vào  $Y$  thì rõ ràng  $W$  là fa-phụ thuộc vào cả  $X$  và  $Y$ . Vậy ta có thể hợp về trái của hai fa-phụ thuộc để tạo thành một fa-phụ thuộc.

**Luật 3.15.** Tối thiểu. Cho  $X, Y, W$  là 3 tập khác rỗng. Nếu  $XY \xrightarrow{fa} W$  và  $Y \xrightarrow{fa} W$ , không tồn tại  $X \xrightarrow{fa} W$  thì  $XY \xrightarrow{fa} W$  được giảm để tạo thành  $Y \xrightarrow{fa} W$ .

Nếu  $W$  chỉ phụ thuộc vào một phần các thuộc tính về trái của luật thì về trái của phụ thuộc phải được giảm còn tập con các thuộc tính mà  $W$  phụ thuộc vào. Vậy thì  $X$  phải được loại bỏ ra khỏi về trái của quan hệ.

**Luật 3.16.** Giảm. Cho  $X, Y, W$  là 3 tập khác rỗng. Nếu  $XYW \xrightarrow{fa} V, Y \xrightarrow{fa} W$  thì  $XYW \xrightarrow{fa} V$  được giảm thành  $XY \xrightarrow{fa} V$ .

Nếu một số thuộc tính trên về trái của một fa-phụ thuộc, chính chúng cũng tạo ra một fa-phụ thuộc thứ cấp thì về trái của phụ thuộc ban đầu cần loại bỏ các thuộc tính giống các thuộc tính ở về phải phụ thuộc thứ cấp.

#### 4. QUÁ TRÌNH PHÁT HIỆN CÁC LỚP BỘ PHẬN MỜ

Dựa vào định nghĩa các dạng chuẩn đối tượng trong [9], tác giả đề xuất qui trình xác định các lớp bộ phận bao gồm các bước sau:

*Bước 1:* Xác định các fa-phụ thuộc.

*Bước 2:* Kiểm tra các fa-phụ thuộc bằng cách sử dụng các luật dẫn xuất.

*Bước 3:* Xác định, kiểm tra các  $fm$ -phụ thuộc.

*Bước 4:* Dựa vào định nghĩa 3.4, xác định ký hiệu ngữ nghĩa của các lớp bộ phận.

*Bước 5:* Xác định biên các lớp bộ phận.

Việc xác định các fa-phụ thuộc trong lược đồ khái niệm dựa vào cả sự hiểu biết về ứng dụng và phân tích ngữ nghĩa các thuộc tính của lớp. Các luật 3.1 đến luật 3.8 và luật 3.13 có thể giúp ta phát hiện ra các fa-phụ thuộc cần thêm vào và loại bỏ các fa-phụ thuộc yếu.

Việc xác định  $fm$ -phụ thuộc có thể được thực hiện bằng cách, với mỗi phương thức của lớp ta quét mỗi câu lệnh trong phương thức, một thuộc tính được thêm vào tập thuộc tính nếu thuộc tính này xuất hiện trong câu lệnh. Khi tất cả các câu lệnh của phương thức được phân tích,  $fm$ -phụ thuộc giữa tập thuộc tính và phương thức được tạo thành. Sau đó ta sử dụng các luật 3.9, 3.10, 3.11, 3.12, 3.13 để kiểm tra các  $fm$ -phụ thuộc được sinh ra.

Sau khi tất cả các fa-phụ thuộc và  $fm$ -phụ thuộc được thiết lập một cách chính xác, vé trái của mỗi fa-phụ thuộc tạo thành ký hiệu ngữ nghĩa của lớp bộ phận tương ứng. Một biên lớp bộ phận là tập hợp tất cả các fa-phụ thuộc mà có cùng ký hiệu ngữ nghĩa  $S$ , tập các thuộc tính và các phương thức sử dụng các thuộc tính đó. Các luật 3.14, 3.15, 3.16 giúp cấu trúc lại các phụ thuộc, nối rộng hay giảm các ký hiệu ngữ nghĩa  $S$ . Trong một lớp đối tượng mờ nếu xuất hiện hơn một lớp bộ phận có thể dẫn đến dư thừa dữ liệu. Do đó, để hạn chế sự dư thừa dữ liệu trong lớp chúng ta phải phân rã lớp đó thành các lớp chỉ chứa một lớp bộ phận (chúng tôi trình bày trong bài báo sau).

Áp dụng qui trình xác định các lớp bộ phận trong lớp Kiemke, ngoài lớp bộ phận Nhakho còn có các lớp bộ phận khác sau:

*Lớp bộ phận Hang*

$$\begin{aligned} &\{\text{masohang, tinhtrang, cuahang, soluong, ngaynhap}\} \xrightarrow{f_a} \{\text{chiphi, giabanle}\} \\ &\{\text{masohang, tinhtrang, cuahang, soluong, ngaynhap}\} \xrightarrow{f_a} \{\text{masokho, dckhohang}\} \\ &\{\text{chiphi, giabanle}\} \xrightarrow{f_m} \{\text{suagiaban}(\dots)\} \\ &\{\text{masohang, tinhtrang, ngaynhap, cuuhang, giabanle, chiphi}\} \xrightarrow{f_m} \{\text{nhaphang}(\dots)\} \\ &\{\text{cuahang}\} \xrightarrow{f_m} \{\text{baocaokke}(\dots)\} \\ &\{\text{masohang, soluong, ngaynhap, masokho, dckhohang}\} \xrightarrow{f_m} \{\text{thongkenhap}(\ )\} \\ &\{\text{masohang, tinhtrang, cuahang, soluong, masokho, dckhohang}\} \xrightarrow{f_m} \{\text{thongkekho}(\ )\} \end{aligned}$$

*Lớp bộ phận Cuahangcc*

$$\begin{aligned} &\{\text{cuahang}\} \xrightarrow{f_a} \{\text{nhanvien}\} \\ &\{\text{nhanvien}\} \xrightarrow{f_m} \{\text{suanhanvien}(\dots)\} \end{aligned}$$

$$\{\text{cuahang}\} \xrightarrow{fm} \{\text{suacuahang}(\dots)\}$$

## 5. KẾT LUẬN

Các thuộc tính trong bất kỳ cơ sở dữ liệu nào đều có mối liên kết, ràng buộc với nhau. Riêng trong các cơ sở dữ liệu hướng đối tượng còn có mối quan hệ giữa các thuộc tính và phương thức, trên cơ sở đó chúng tôi định nghĩa hai kiểu phụ thuộc mờ đó là fa-phụ thuộc và  $fm$ -phụ thuộc và các luật suy diễn trên các phụ thuộc đó. Trên cơ sở đó phát hiện các lớp bộ phận tồn tại trong cùng một lớp đối tượng mờ, là nguyên nhân tiềm ẩn dư thừa dữ liệu. Để tối ưu hoá một lược đồ cơ sở dữ liệu hướng đối tượng mờ, một bước quan trọng tiếp theo là chuẩn hoá các lớp trong lược đồ cơ sở dữ liệu hướng đối tượng mờ. Các dạng chuẩn lớp mờ và phương pháp tách lớp về các dạng chuẩn cũng như giải thuật để phát hiện các lớp bộ phận tồn tại trong lớp đối tượng mờ sẽ được chúng tôi trình bày trong các nghiên cứu tiếp theo.

## TÀI LIỆU THAM KHẢO

- [1] B.Bouchon-Meunier, Hồ Thuần, Đặng Thanh Hà, Logic mờ và Ứng dụng, Đại học Quốc gia Hà Nội, 2007.
- [2] Byung S.Lee, Normalization in OODB Design, *ACM SIGMOD Record*, Vol 24 (3) - 1995, 23-27.
- [3] Catriel Beeri, Ronald Fagin, and John H. Howard, A complete axiomatization for functional and multivalued dependencies in database relations, *Proceedings of ACM SIGMOD*, August, 1997, 47-63.
- [4] Đặng Thanh Hà, Về phép tách và phép kết nối trong quan hệ có thuộc tính mờ, *Tạp chí Tin học và Điều khiển học*, 2006, 183-186.
- [5] Hồ Thuần, Hồ Cẩm Hà, Cơ sở dữ liệu Lý thuyết và Thực hành (tập 2), Nhà xuất bản giáo dục, 2004.
- [6] K. v. s. v. n. Raju, Arun k. Majumdar, Fuzzy Functional Dependencies and Lossless Join Decomposition of Fuzzy Relational Database Systems, *ACM Transactions on Database Systems*, Vol 13 (2) - 1988, 129 - 166.
- [7] Nguyễn Xuân Huy, Các phụ thuộc logic trong cơ sở dữ liệu, Nhà xuất bản Thống kê - Hà Nội, 2006.
- [8] Sozat M I, Yazici A, A complete axiomatization for fuzzy functional and multivalued dependencies in fuzzy database relations, *Fuzzy Sets and Systems*, 2001, 161-183.
- [9] Tarek Sobh, *Advances in Computer and Information Sciences and Engineering*, Springer, NewYork, 2008, 300-304.
- [10] Zadeh, L. A., Fuzzy sets, *Information and Control*, Vol 8 (3) - 1965, 338-353.
- [11] Zongmin Ma, *Fuzzy Database Modeling with XLM*, Springer, NewYork, 2005.

Nhận bài ngày 21 - 7 - 2010

Nhận lại sau sửa ngày 5 - 11 - 2010