

# MỘT TIẾP CẬN CHUẨN HÓA CƠ SỞ DỮ LIỆU HƯỚNG ĐỐI TƯỢNG

• Nguyễn Hữu Duyệt (\*)

## TÓM TẮT

Với sự phát triển nhanh chóng của công nghệ thông tin, mô hình quan hệ cũng đã bộc lộ một số hạn chế về khả năng lưu trữ trên các đối tượng dữ liệu phức tạp. Trong khi đó mô hình dữ liệu hướng đối tượng hứa hẹn nhiều khả năng cho việc mô hình hóa thế giới thực ngày càng yêu cầu sự biểu diễn phức tạp. Nhưng mô hình dữ liệu hướng đối tượng cũng đang đứng trước những thách thức, bao gồm những vấn đề: chưa có một chuẩn chính thức được chấp nhận bởi các tổ chức ANSI hoặc ISO như đối với mô hình quan hệ [2][3][5]; các nền tảng lý thuyết về cơ sở dữ liệu hướng đối tượng (OODB) cũng đang trong quá trình nghiên cứu và phát triển. Bài báo này có mục đích nghiên cứu một phương pháp tiếp cận để chuẩn hóa đối tượng, dựa vào các khái niệm về ràng buộc phụ thuộc đường đi trong mô hình đối tượng, xuất phát từ một công trình nghiên cứu của Tari, Stokes và Spaccapietra [7].

### 1. Đặt vấn đề

Trong mô hình quan hệ, việc chuẩn hóa các lược đồ quan hệ đã có một hệ thống cơ sở lý thuyết vững chắc. Nhưng trong mô hình OODB, hiện tại chưa xây dựng được các phương pháp chuẩn hóa dữ liệu một cách đầy đủ có căn cứ khoa học, bởi những khó khăn là lược đồ hướng đối tượng có những đặc điểm khác với lược đồ quan hệ. Một lược đồ không ở dạng chuẩn 1NF trong lược đồ quan hệ, nhưng có thể tương ứng với một lược đồ hướng đối tượng chấp nhận được. Việc phân giải lược đồ quan hệ về dạng chuẩn 2NF, 3NF có thể được vận dụng trong thiết kế OODB. Tuy nhiên các vấn đề dị thường cập nhật vẫn chưa được giải quyết và hiệu năng của hệ thống cũng là một vấn đề đặt ra.

### 2. Một hướng tiếp cận chuẩn hóa trong mô hình dữ liệu hướng đối tượng

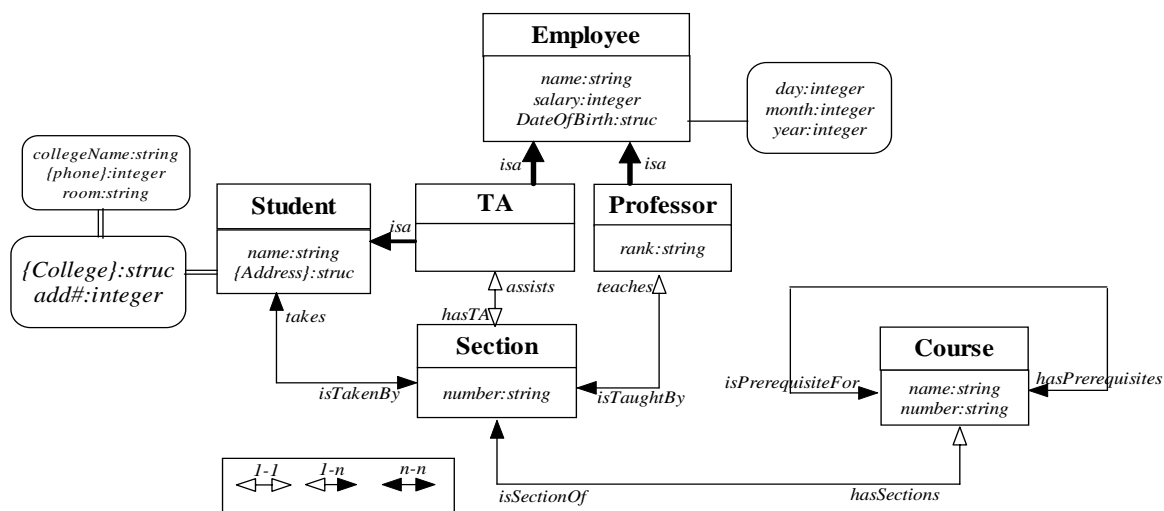
#### 2.1. Một số nhược điểm của các tiếp cận thiết kế truyền thống

(1) Mô hình dữ liệu không đủ mạnh để hỗ trợ các cơ chế mô hình hóa phức hợp (như định danh đối tượng, các đối tượng phức hợp và tính kế thừa).

(2) Các ràng buộc phụ thuộc trong các mô hình quan hệ phẳng và lồng nhau có thể được xếp vào loại ràng buộc dựa trên giá trị, nghĩa là chúng phản ánh mối quan hệ phụ thuộc giữa các dữ liệu trong một quan hệ [2].

(\*) ThS. Trưởng Khoa CNTT, Trường Đại học Đồng Tháp

**Hình 2.1.** Lược đồ dữ liệu OO cho quản lí các khóa học



**2.2. Một số phép toán đại số đối tượng**

**2.2.1. Mô hình dữ liệu và vấn đề ràng buộc**

Trong phần này trình bày các phép toán trên đối tượng, các phép toán trên giá trị bộ, các phép toán trên giá trị tập, các phép chiếu đối tượng.

*Bảng 2.1. Một thể hiện của lược đồ OODB trong Hình 2.1.*

<b>Employee</b>						
Id <sub>Emp</sub>	Name	Salary	DateOfBirth			
			bth#	day	Month	Year
E1	Hoàng	1.500.000	1	10	12	1960
E2	Lan	1.800.000	2	13	05	1962
E3	Minh	1.300.000	3	03	07	1980

<b>Student</b>							
Id <sub>Stu</sub>	name	Address					{takes}
		{College}					
		col#	collegeName	{phone}	room	add#	
S1	Hoa	1	Hà Nội 1	04.8261472	P1	1	SE1
				04.8261473			SE2
		2	HCM	08.8816573	P3	2	SE3

S2	Hùng	1	Hà Nội 1	04.8261472	P2	1	SE1
		4	Vinh	04.8261473 038.856734	P2	3	SE2
S3	Hoa	2	HCM	08.8816573	P1	2	SE2
				08.8816574			SE3
S4	Dung	2	HCM	08.8816573	P2	2	SE1
				08.8816574			

<i>TA</i>	
Id <sub>TA</sub>	Assist
E2	SE1
S1	SE3

<i>Professor</i>		
Id <sub>Pro</sub>	Rank	{teaches}
E1	Hight	SE1
		SE3
E3	Hight	SE2
		SE3

<i>Section</i>					
Id <sub>Sec</sub>	number	isSectionOf	Assist	{isTakenBy}	isTaughBy
SE1	1	C1	E2	S2	E1
				S3	
SE2	2	C1	E2	S3	E3
				S4	
SE3	3	C2	S1	S2	E1

<i>Course</i>				
<b>Id<sub>C<sub>o</sub></sub></b>	<b>Name</b>	<b>Number</b>	<b>{hasPrerequisites}</b>	<b>{isPrerequisites}</b>
C1	Hệ ĐH	1	Null	2 3
C2	Mạng MT	2	2	3
C3	Internet	3	1 2	Null

Một đối tượng  $O$ , được biểu diễn như là  $O(\Delta_o)$ , với  $\Delta_o$  là kiểu đối tượng. Ví dụ, trong **Hình 2.1** đối tượng *Professor* được định nghĩa như  $Professor([Id_{Emp}: allID; name: string; salary: integer; DateOfBirth: struc; rank: string; \{teaches: Section\}])$ , trong đó:

- Đối tượng *DateOfBirth* được định nghĩa như  $DayOfBirth([day: integer; month: integer; year: integer])$ ;
- *allID* miền của tất cả các định danh đối tượng;
- Thuộc tính *DateOfBirth* là thuộc tính phức hợp của ba thuộc tính đơn *day*, *month*, *year*;
- Thuộc tính *teaches* mô hình hóa ngữ nghĩa mối quan hệ kết hợp giữa đối tượng *Professor* và đối tượng *Section*.

Một thể hiện của *Professor* được xác định bởi  $E1$  và có dạng  $[E1, Hoàng, 1.500.000, [1, 10, 12, 1960], hight, \{SE1, SE3\}]$ .

### 2.2.2. Các khái niệm về đường đi

Tari [7] đã phát triển các đại số phép chiếu dựa trên khái niệm đường đi.

#### ▪ Đường đi

**Định nghĩa 2.1:** Một đường đi từ một đối tượng  $O_1$  tới đối tượng  $O_n$  ( $n \geq 2$ ) thông qua các đối tượng  $O_2, \dots, O_{n-1}$  được kí hiệu  $O_1-O_2-\dots-O_n$ . Có ba kiểu đường đi.

#### ▪ Đường đi dọc

**Định nghĩa 2.2:** Đường đi dọc là một đường đi mà dãy đối tượng bắt đầu từ một đối tượng tới một trong những thành phần trực tiếp của nó.

#### ▪ Đường đi ngang

**Định nghĩa 2.3:** Là một đường đi mà dãy các đối tượng chỉ có các đối tượng (thực thể) đan xen nhau.

#### ▪ Đường đi hỗn hợp

*Định nghĩa 2.4:* Là một đường đi mà dãy các đối tượng là một tổ hợp của các đường đi ngang và dọc.

▪ *Thể hiện đường đi*

*Định nghĩa 2.5:* Một thể hiện của đường đi  $O_1-O_2-...-O_n$  là một dãy các thể hiện có dạng  $\tau_1-\tau_2-...-\tau_n$  với  $\tau_i$  là một thể hiện của  $O_i$ ,  $i = 1, 2, \dots, n$ .

2.2.3. *Các phép chiếu theo đường đi*

▪ *Khái niệm phép chiếu theo đường đi*

Một phép chiếu theo một đường đi hoặc một thể hiện đường đi có thể được định nghĩa như là một sự lồng nhau của nhiều phép chiếu trên các thành phần của một đối tượng. Cho một thể hiện đối tượng  $\tau$  và một đường đi hoặc một thể hiện đường đi  $\rho$ , kí hiệu:

- $\tau[\rho]$  là phép chiếu của  $\tau$  lên đối tượng  $O$ ;
- $\tau_\rho[\rho]$  là phép chiếu của  $\tau$  lên đối tượng  $O$  theo đường đi  $\rho$ .

Chú ý rằng, một cách ngắn gọn, có thể dùng định danh để chỉ một thể hiện đối tượng. Chẳng hạn, dùng *SI* để chỉ  $[SI, Hoa, \{[1, Hà Nội1, \{04.8261472, 04.6261473\}, P1, 1], [2, HCM, \{08.865723, 08.865724\}, P2, 2], [3, Hà Nội 2, \{04.8832614\}, P2, 1]\}]$  của *Student*.

▪ *Phép chiếu đơn giản*

*Định nghĩa 2.6:* Một phép chiếu đơn giản là phép chiếu chỉ liên quan đến một đường đi ngang hoặc dọc một cấp.

Một cách hình thức, phép chiếu đơn giản của một thể hiện  $\tau = [\tau_1, \dots, \tau_j, \dots, \tau_n]$  của đối tượng  $O([O_1, \dots, O_j, \dots, O_n])$  lên thành phần  $O_j$  trực tiếp của nó được định nghĩa như là  $\tau[O_j] = \{\tau_j\}$ .

▪ *Phép chiếu phức hợp*

*Định nghĩa 2.7:* Phép chiếu phức hợp là phép chiếu có liên quan đến đường đi nhiều cấp.

Một cách hình thức, phép chiếu của thể hiện  $\tau$  của đối tượng  $O$  lên đối tượng khác  $O_q$  theo đường đi hỗn hợp  $O-O_1-...-O_{q-1}$  được định nghĩa bằng cách chiếu liên tiếp lên các đối tượng  $O_1, \dots, O_{q-1}$ , hay ta viết  $\tau[O_q] = (\tau[O_{q-1}])[O_q] = (\dots(\tau[O_1])[O_2])\dots[O_{q-1}][O_q]$ .

▪ *Phép chiếu điều kiện đơn giản*

*Định nghĩa 2.8:* Phép chiếu điều kiện của một thể hiện  $\tau$  của đối tượng  $O$  trên đối tượng thành phần trực tiếp  $O_i$  theo thể hiện đường đi  $\tau_1-\tau_2$  của đường đi  $O-O_i$  là

$$\text{Nếu } \tau \neq \tau_1 \text{ thì } \tau_{\tau_1-\tau_2} = \emptyset$$

$$\text{Ngược lại nếu } O_i \text{ là một đối tượng đơn trị thì } \tau_{\tau_1-\tau_2} = \tau_2$$

$$\text{Ngược lại } \tau_{\tau_1-\tau_2} = \{y \mid y \in \tau[O_i] \text{ và } y = \tau_2\}.$$

▪ *Phép chiếu điều kiện phức hợp*

*Định nghĩa 2.9:* Phép chiếu của một thể hiện  $\tau$  của đối tượng  $O$  lên một đối tượng khác  $O_p$  theo thể hiện đường đi  $\tau-\tau_1-\dots-\tau_{p-1}-\tau_p$  của một đường đi hỗn hợp  $O-O_1-\dots-O_{p-1}-O_p$  là

$$\tau_{\tau-\tau_1-\dots-\tau_{p-1}-\tau_p}[O_p] = (((\tau_{\tau-\tau_1}[O_1])\dots)_{\tau_{p-2}\tau_{p-1}\tau}[O_{p-1}])_{\tau_{p-1}-\tau_p}[O_p].$$

Việc làm phẳng một thể hiện  $\tau = [\tau_1, \dots, \tau_j, \dots, \tau_n]$  của đối tượng  $O([O_1, \dots, \{O_j\}, \dots, O_n])$  lên một thành phần đa trị trực tiếp của nó  $O_j$  được định nghĩa là  $\tau[O_j] = \cup_k t_k, t_k \in \tau[\{O_j\}]$ .

**2.3. Các ràng buộc phụ thuộc**

**2.3.1. Các ràng buộc phụ thuộc đường đi**

Ta gọi kí hiệu tất cả các thể hiện của một CSDL là  $\psi$ . Phép chiếu của tập hợp này lên đối tượng  $X$ , kí hiệu là  $\psi[X]$ , biểu diễn tất cả các thể hiện của đối tượng đó.  $\psi[X]$  còn gọi là một thể hiện thế giới của  $X$ .

*Định nghĩa 2.10. (Ràng buộc-PD)* Một phụ thuộc đường đi giữa đối tượng  $X$  và  $Y$  theo đường đi  $\rho$ , kí hiệu  $X \xrightarrow{\rho} Y \Leftrightarrow \forall x_0 \in \psi[X], \forall \rho_1, \forall \rho_2 \in \psi[\rho]$  thì  $x_{0_{\rho_1}}[Y] = x_{0_{\rho_2}}[Y]$ .

**2.3.2. Các ràng buộc phụ thuộc cục bộ**

*Định nghĩa 2.11. (Ràng buộc - LD)* Một đối tượng  $Y$  được gọi là phụ thuộc cục bộ vào đối tượng  $X$ , kí hiệu  $X \rightarrow Y \Leftrightarrow \forall x_0 \in \psi[X], \rho_1, \rho_2$  là hai đường đi khác nhau từ  $X$  tới  $Y$  và  $\tau_1 \in \psi[\rho_1], \tau_2 \in \psi[\rho_2]$  chứa  $x_0$  thì  $x_{0_{\tau_1}}[Y] = x_{0_{\tau_2}}[Y]$ .

**2.3.3. Các ràng buộc phụ thuộc toàn cục**

*Định nghĩa 2.12: (Ràng buộc-GD).* Một đối tượng  $Y$  được gọi là phụ thuộc toàn cục vào đối tượng  $X$  (viết tắt là  $Y$  phụ thuộc vào  $X$ ), kí hiệu  $X \rightarrow Y \Leftrightarrow X \rightarrow Y$  và  $\forall x_1, x_2 \in \psi[X], x_1 = x_2, \forall \rho_1, \rho_2, \forall \tau_1, \tau_2$  với  $\rho_1, \rho_2$  là các đường đi chứa  $X$  và  $Y$  và  $\tau_1, \tau_2$  là hai thể hiện đường đi tương ứng của  $\rho_1, \rho_2$  chứa các thể hiện tương ứng  $x_1, x_2$  thì  $x_{1_{\tau_1}}[Y] = x_{2_{\tau_2}}[Y]$ .

Nếu  $X$  là một đối tượng thực thể thì

$$\psi[X] = \{ \tau \mid \tau \in \psi, \tau \text{ là một thể hiện của } X \} \subseteq \psi.$$

Ngược lại, nếu  $X$  là một thuộc tính thì

$$\psi[X] = \{ y \mid \exists E, \forall \tau, \tau \text{ là một thể hiện của } E, E \text{ là cha của } X, \text{ và } y \in \tau_{path_X(E)}[X] \} \subseteq \psi.$$

Với  $path_X(E)$  biểu diễn đường đi ngắn nhất giữa  $E$  và  $X$ .

Cho đối tượng  $O$ , ta kí hiệu  $path_{O_1O_2...O_n}(O)$  là đường đi ngắn nhất xuất phát từ  $O$  và chứa tất cả các đối tượng  $O_1, O_2, \dots, O_n$ . Như vậy,  $path_{O_1O_2...O_n}(O)$  thỏa mãn các tính chất:

- $path_{O_1O_2...O_n}(O)$  chứa  $O, O_1, O_2, \dots, O_n$ ;
- $\forall \rho_{O_1O_2...O_n}$ , với  $\rho_{O_1O_2...O_n}$  là một đường đi xuất phát từ  $O$  và chứa tất cả các đối tượng  $O_1, O_2, \dots, O_n$  sao cho  $length(path_{O_1O_2...O_n}(O)) \leq length(\rho_{O_1O_2...O_n})$ . Trong đó hàm  $length(\text{đường đi})$  trả về số các thuộc tính thuộc đường đi.

#### 2.3.3.4. Ràng buộc khóa

**Định nghĩa 2.13:** (Ràng buộc khóa)  $K$  là một khóa ứng viên của đối tượng  $O$  nếu và chỉ nếu:

- (1)  $K$  là một tập con của tập các thành phần của đối tượng  $O$ ;
- (2)  $\exists S_1, S_2$  sao cho:
  - (a)  $S_1, S_2$  và  $K$  tạo thành một phân hoạch của  $O$ ;
  - (b)  $\forall O_i \in S_1, K \rightarrow O_i$ , và;
  - (c)  $\forall O_j \in S_2, \exists K_{A_1}, K_{A_2}, \dots, K_{A_r}$  sao cho  $K_{A_i}$  là một khóa của  $A_i$  và  $A_i$  là một đối tượng cha của  $O$  ( $1 \leq i < r$ ), và  $K \cup K_{A_1} K_{A_2} \dots K_{A_r} \rightarrow O_j$ .
- (3) Không có tập con nào thực sự của  $K$  thỏa mãn điều kiện (1) và (2).

### 2.4. Chuẩn hoá lược đồ dữ liệu hướng đối tượng

#### 2.4.1. Mô hình của một đối tượng

##### 2.4.1.1. Các khái niệm

**Định nghĩa 2.13.** (Phụ thuộc cố kết – coherent dependency) Một phụ thuộc-GD  $A \rightarrow B$  được gọi là phụ thuộc cố kết nếu và chỉ nếu  $A-B$  là một đường đi; ngược lại nó được gọi là phụ thuộc không cố kết [7].

**Định nghĩa 2.14.** (Thể hiện đối tượng) Một thể hiện của một đối tượng là một tập các ràng buộc-GD cố kết của đối tượng đó. Một thể hiện đối tượng được gọi là thể hiện tối đại nếu và chỉ nếu nó không thể được dẫn xuất từ bất kỳ một thể hiện đối tượng nào khác [4][6].

**Định nghĩa 2.15.** (Mô hình) Cho  $I_1, I_2, \dots, I_n$  là tất cả các thể hiện tối đại của một đối tượng  $O$ , thì thể hiện  $I_1 \vee I_2 \vee \dots \vee I_n$  được gọi là mô hình của đối tượng  $O$ .

### 2.4.1.2. Định lý về điều kiện cần và đủ của mô hình đối tượng [7]

**Định lý 2.1.** *M là mô hình của đối tượng E nếu và chỉ nếu M thỏa mãn các điều kiện sau:*

(1) Nếu  $Id_E$  là định danh của đối tượng E thì:

(a) Nếu A là một thuộc tính đơn trị trực tiếp của E thì  $Id_E \rightarrow A \in M$ ;

(b) Nếu A là một thuộc tính đa trị trực tiếp của E thì  $Id_E \rightarrow \{A\} \in M$ ;

(c) Nếu K là một khóa của E và K không là định danh của E thì  $K \rightarrow Id_E \in M$  và  $Id_E \rightarrow K \in M$ .

(2) Nếu  $A_1$  là một thuộc tính phức hợp của E,  $Id_{A_1}$  là định danh của  $A_1$ , và

(a) B là một thuộc tính đơn trị trực tiếp của  $A_1$  thì  $Id_{A_1} \rightarrow B \in M$  và  $Id_E Id_{A_1} \rightarrow B \in M$ ;

(b) B là một thuộc tính đa trị trực tiếp của  $A_1$  thì  $Id_{A_1} \rightarrow \{B\} \in M$  và  $Id_E Id_{A_1} \rightarrow \{B\} \in M$ ;

(c)  $K_{A_1}$  là một khóa của  $A_1$  và không là định danh của  $A_1$  thì  $K_{A_1} \rightarrow Id_{A_1} \in M$  và  $Id_{A_1} \rightarrow K_{A_1} \in M$ .

(3) Nếu  $A_1, \dots, A_n$  là các thành phần của E, với  $A_1 - \dots - A_n$  là một đường đi của E,  $Id_{A_i}$  là định danh của  $A_i$  ( $1 \leq i \leq n$ ), và

(a) Nếu B là một thuộc tính đơn trị của  $A_n$ , thì  $\forall j, 0 \leq j \leq n$   $Id_{A_j} \dots Id_{A_{n-1}} Id_{A_n} \rightarrow B \in M$ , với  $Id_{A_0} = Id_E$ ;

(b) Nếu B là một thuộc tính đa trị trực tiếp của  $A_n$ , thì  $\forall j, 0 \leq j \leq n$   $Id_{A_j} \dots Id_{A_{n-1}} Id_{A_n} \rightarrow \{B\} \in M$ , với  $Id_{A_0} = Id_E$ ;

(c)  $K_{A_n}$  là một khóa của  $A_n$  và không là định danh của  $A_n$ , thì  $K_{A_n} \rightarrow Id_{A_n} \in M$  và  $Id_{A_n} \rightarrow K_{A_n} \in M$ .

(4) Ngoài các phụ thuộc dạng (1), (2), (3) và các phụ thuộc được suy diễn từ chúng nhờ các tiên đề suy diễn, không có một phụ thuộc nào khác thuộc M.

### 2.4.3. Các dạng chuẩn đối tượng

#### 2.4.3.1. Khái niệm dạng chuẩn đối tượng

**Định nghĩa 2.16.** (Dạng chuẩn đối tượng) Một đối tượng E được gọi là thuộc một dạng chuẩn, nếu và chỉ nếu:

(i) Mỗi thuộc tính phức hợp của E có một định danh;

(ii)  $\sum_E \mid = \sum_{user}^E$ .

#### 2.4.3.2. Định lý về điều kiện cần cho dạng chuẩn

**Định lý 2.2.** Nếu một đối tượng E thuộc một dạng chuẩn thì các mệnh đề sau thỏa mãn:

(1) Nếu  $A \rightarrow B \in \sum_{user}^E$ , với A, B là các thành phần của E, thì  $A \rightarrow B \in \sum_E$ .



(2) Nếu  $B$  là một thành phần đơn trị trực tiếp của đối tượng  $O_n$  thuộc  $E$ , thì tồn tại một tập các thuộc tính khóa  $K_{O_1}, \dots, K_{O_n}$ , với  $K_{O_i}$  là khóa của  $O_i$  thuộc  $E$  và  $O_{i+1}$  là thành phần trực tiếp của  $O_i$  ( $1 \leq i \leq n-1$ ) sao cho  $K_{O_1} \dots K_{O_n} \rightarrow B \in \Sigma_{user}^E$ .

(3) Nếu  $B$  là một thành phần đa trị trực tiếp của đối tượng  $O_n$  thuộc  $E$ , thì tồn tại một tập các thuộc tính khóa  $K_{O_1}, \dots, K_{O_n}$ , với  $K_{O_i}$  là khóa của  $O_i$  thuộc  $E$  và  $O_{i+1}$  là thành phần trực tiếp của  $O_i$  ( $1 \leq i \leq n-1$ ) sao cho  $K_{O_1} \dots K_{O_n} \rightarrow \{B\} \in \Sigma_{user}^E$ .

(4) Không có phụ thuộc nào đối với các thuộc tính không khóa của một thành phần  $O$  của  $E$ .

#### 2.4.3.3. Định lý về điều kiện cần và đủ cho dạng chuẩn

**Định lý 2.3.** Một đối tượng  $E$  thuộc một dạng chuẩn nếu và chỉ nếu:

(1) Nếu  $A \rightarrow B$  là một phụ thuộc của người sử dụng, thì phải thỏa một trong các điều kiện:

(a)  $A-B$  là một đường đi của  $E$ ;

(b)  $A$  là một tập các khóa của  $O_1, \dots, O_n$  là các thành phần của  $E$ , sao cho  $O_{i+1}$  là thuộc tính trực tiếp của  $O_i$  ( $1 \leq i \leq n-1$ ), và  $B$  là một thành phần đơn trị trực tiếp của  $O_n$ .

(c)  $A$  là một thành phần trực tiếp của đối tượng  $O$  với bản số 1:n và  $B$  là một thành phần đơn trị trực tiếp của  $O$ .

(2) Nếu  $A \rightarrow \{B\}$  là một phụ thuộc của người sử dụng thì phải thỏa một trong các điều kiện:

(a)  $A-B$  là một đường đi của  $E$ ;

(b)  $A$  là một tập các khóa của  $O_1, \dots, O_n$  là các thành phần của  $E$ , sao cho  $O_{i+1}$  là thuộc tính trực tiếp của  $O_i$  ( $1 \leq i \leq n-1$ ), và  $B$  là một thành phần đa trị trực tiếp của  $O_n$ .

(c)  $A$  là một thành phần trực tiếp của đối tượng  $O$  với bản số 1:n và  $B$  là một thành phần đơn trị trực tiếp của  $O$ .

#### 2.4.4. Các qui tắc chuẩn hóa đối tượng

**Qui tắc 2.1.** Cho đối tượng  $E$ , nếu có một phụ thuộc không đồng nhất  $A \rightarrow B$  hoặc  $A \rightarrow \{B\}$  nhưng có một đường đi chứa  $A$  và  $B$ , thì  $path_B(A)$  được chia thành hai đường đi con  $\rho_1, \rho_2$  sao cho  $\rho_1 = A-B, \rho_2 = path_B(A) \setminus B$ .

**Qui tắc 2.2.** Cho đối tượng  $E$ , nếu tất cả các phụ thuộc của  $E$  có các đối tượng phức hợp là những phụ thuộc đồng nhất, và nếu có một phụ thuộc-GD  $A \rightarrow B$  không suy dẫn được từ  $\Sigma_{user}^E$ , thì  $E$  được biến đổi bằng cách tạo nên một đối tượng mới  $O_{new}$  có những tính chất sau:

- $O_{new}$  là thành phần trực tiếp của  $E$  có bản số  $(\min(k_{o_1}, \dots, k_{o_n}), \min(K_{o_1}, \dots, K_{o_n}))$ , với  $A = O_1 \dots O_n$  và  $(k_{o_j}, K_{o_j})$  là các bản số cực tiểu và cực đại của thuộc tính  $O_j$ . Thuộc tính  $O_j$  trở thành một thành phần đơn trị trực tiếp của  $O_{new}$  ( $1 \leq j \leq n$ );

- $B$  là một thành phần trực tiếp của  $O_{new}$ ;
- $A$  là một khóa của  $O_{new}$ .

Tari [7] đã chứng tỏ dùng Qui tắc 2.1 và Qui tắc 2.2 là đủ để biến đổi bất kì một đối tượng nào về dạng chuẩn.

#### 2.4.5. Sinh ra các dạng chuẩn bằng việc xây dựng lại

Có 3 giải thuật sau:

Giải thuật 1 (*DECOMPOSE()*) thực hiện việc phân giải đối tượng dựa trên tập các ràng buộc-GD của người sử dụng thành các cụm không thể phân giải được nữa.

Giải thuật 2 (*BUID\_TREE()*) xây dựng cấu trúc cây cho mỗi cụm bằng cách sử dụng khoá xác định cho cụm cũng như ràng buộc-GD có liên quan đến các đối tượng trong cụm. Bước này xây dựng các cấu trúc phù hợp của đối tượng bằng việc xác định mối quan hệ giữa các cụm thông qua các khóa của các cụm. Nói cách khác, nếu khóa  $K_1$  được bao hàm trong khóa khác  $K_2$ , thì cây của cụm xác định bởi  $K_2$ .

Giải thuật 3 (*MERGER()*) trộn các cây này theo các nút và các cạnh chung của chúng, tạo nên các cấu trúc phức hợp cho đối tượng chuẩn hóa.

##### 2.4.5.1. Các giải thuật phân giải và trộn (decompose and fusion algorithms)

a) **Định nghĩa 2.18:** Cho tập hợp các ràng buộc-GD  $S$ , tập hợp con tối thiểu chứa tất cả các ràng buộc có thể dẫn xuất từ  $S$  sử dụng các tiên đề suy diễn gọi là bao đóng của  $S$ , kí hiệu  $S^{cover}$ .

$S^{cover}$  có thể chứa những ràng buộc-GD dư thừa hoặc có thể dẫn xuất từ các ràng buộc khác, vì vậy ta có thể xoá đi tất cả các ràng buộc-GD này khỏi tập  $S^{cover}$ , để sinh ra một tập mới gọi là phủ tối thiểu của  $S$ , kí hiệu  $S^{minimal}$ .

b) **Định nghĩa 2.19 (Ràng buộc-GD dư thừa).** Cho  $S$  là tập hợp các ràng buộc-GD, phụ thuộc  $X \rightarrow Y \in S^{cover}$  được gọi là

- Tầm thường, nếu  $Y = \emptyset$  hoặc  $Y \subseteq X$ ;
- Khả rút gọn trái, nếu  $\exists X', X' \subseteq X$  và  $X' \rightarrow Y \in S^{cover}$ .

c) *Định nghĩa 2.20*: Một phụ thuộc được gọi là ở dạng rút gọn nếu nó không là phụ thuộc tầm thường hoặc khả rút gọn trái.

d) *Định nghĩa 2.21 (Phủ tối thiểu)*. Một tập các ràng buộc-GD  $S$  là một phủ tối thiểu nếu:

-  $\forall w \in S$ ,  $w$  là một ràng buộc ở dạng rút gọn;

-  $\neg \exists S', S' \subset S$  và  $S'^{cover} = S^{cover}$ .

Để tiện cho việc phát biểu giải thuật, ta đưa thêm các khái niệm sau:

-  $DEP(X, S) = \{Y \mid X \rightarrow Y \in S\}$ .

-  $\Sigma_T = \{X \rightarrow Y \mid X \rightarrow Y \in \Sigma_{user}^E \text{ và } X \cap T \neq \emptyset \text{ và } Y \subseteq T\}$ .

-  $KEYS(T) = \{X \cap T \mid X \in LHS(\Sigma_T)\}$ , với  $LHS(\Sigma_T)$  là tập hợp các vế trái của các phụ thuộc của  $\Sigma_T$ .

-  $SUB\_TREE(X, \Sigma) = X \cup \{Y \mid Y \in DEP(Z, \Sigma), Z \neq \emptyset, Z \cap X \neq \emptyset, (Z-X) \cap SUB\_TREE(X, \Sigma) = \emptyset\}$

Để chuẩn hoá một lược đồ  $S = \langle \mathcal{O}, \Sigma \rangle$ , ta cần tiến hành những bước sau:

*Bước 1*: Xác định các cụm, nghĩa là tiến hành phân cụm các đối tượng.

*Bước 2*: Xây dựng cấu trúc cây của các cụm xác định ở bước 1.

*Bước 3*: Trộn các cây được tạo ở bước 2, tạo ra cấu trúc phức hợp của đối tượng cuối cùng.

Một cụm  $T$  có thể phân giải thành các cụm con nếu và chỉ nếu  $KEYS(T)$  là khác rỗng. Giả sử  $KEYS(T) = \{K_1, \dots, K_n\}$  thì các cụm con tương ứng là  $T_1, \dots, T_n$  được lấy bởi hàm  $SUB\_TREE(\cdot)$ :  $T_1 = SUB\_TREE(K_1, \Sigma_T)$ , ...,  $T_n = SUB\_TREE(K_n, \Sigma_T)$ .

**Giải thuật 1** (*Giải thuật phân giải*)

**Procedure DECOMPOSE**( $\mathcal{O}, \Sigma$ )

INPUT: Một lược đồ hướng đối tượng  $S = \langle \mathcal{O}, \Sigma \rangle$ .

OUTPUT: Một lược đồ các cụm không thể phân giải  $T$ .

METHOD

$T := \mathcal{O}$ ;

$\Sigma_T := (\Sigma_E)^{minimal}$ ;

repeat

for each  $T_r \in T$  do

- Tính  $\Sigma_{T_r}$ ;

- Tính  $KEYS(T_r)$ ;

- Xoá khoá của  $T_r$  và khoá của cụm cha của  $T_r$  khỏi  $KEYS(T_r)$ ;

- Xoá các khoá không cực đại khỏi  $KEYS(T_r)$  ;

if  $KEY(T_r) \neq \emptyset$  then

for each  $X \in KEYS(T_r)$  do

- Tính  $SUB\_TREE(X, \Sigma_{T_r})$  ;

-  $T := T \cup SUB\_TREE(X, \Sigma_{T_r})$ ;

- Xoá  $X$  khỏi  $KEYS(T_r)$  ;

endfor

endif

endfor

until  $KEYS(T_r) = \emptyset$ ;

Ví dụ : Cho  $T = ABCDEF$ ,  $\Sigma_T = \{A \rightarrow B, B \rightarrow C, BC \rightarrow D, C \rightarrow F\}$ .

**Giải thuật 2** (Xây dựng các cây từ các cụm)

**Procedure BUILD\_TREE()**

INPUT

Cụm T với các đối tượng  $A_1, \dots, A_n$  và tập ràng buộc-GD  $\Sigma_T = \{GD_1, \dots, GD_m\}$ .

OUTPUT

Cây biểu diễn cho cụm T.

METHOD

$G_1 = \{GD_1\}$ ; //Tập các ràng buộc có cùng vế trái với  $GD_1$

...

$G_m = \{GD_m\}$ ; //Tập các ràng buộc có cùng vế trái với  $GD_m$

(1)for each GD in S ( $GD = X \rightarrow Y$ ) do

- Tìm nhóm  $G_k$  thích hợp cho GD chỉ chứa các ràng buộc-GD có vế trái là X;

- Thêm GD vào  $G_k$ .

*Endfor*

(2) Sắp thứ tự  $G_1, \dots, G_k$  thỏa điều kiện nếu  $X_i \subset X_j$ ,  $X_i \in LHS(D_i)$ ,  $X_j \in LHS(D_j)$ ,  $D_i \in G_i$ ,  $D_j \in G_j$  thì  $i < j$ . Giả sử thứ tự đó là  $G_1, \dots, G_k$ ,  $k \leq m$ , nếu nhiều nhóm trùng nhau thì chỉ giữ lại một nhóm.

(3) //Xây dựng cây

*For*  $i=1$  to  $m$  do

*for each* ràng buộc-GD  $D$  thuộc  $G_i$  do

- Tạo các nút trong là  $LHS(D)$ ;

- Tạo các nút ngoài là  $DEP(LHS(D), \Sigma_T)$ ;

- Tạo liên kết giữa các nút trong và ngoài;

*endfor*

*endfor*

**Giải thuật 3** (Giải thuật trộn)

**Procedure MERGER()**

INPUT

Tập các cây  $T_1, \dots, T_k$ ;

OUTPUT

Cây được trộn TREE

METHOD

$TREE := T_1$ ;  $TREE\_SET := \{T_1, \dots, T_k\}$ ;

*Repeat*

*For each*  $T_i$  in  $TREE\_SET$  do

*If*  $T_i \cap TREE \neq \emptyset$  *Then*

$TREE := MERGE\_TREE(T_i, TREE)$ ;

Xóa  $T_i$  khỏi  $TREE\_SET$ ;

*End If*

*Endfor*

*Until*  $TREE\_SET = \emptyset$

### 3. Kết luận

Bài báo trình bày việc xây dựng các dạng chuẩn hướng đối tượng dựa trên các khái niệm ràng buộc phụ thuộc: phụ thuộc đường đi, phụ thuộc cục bộ, và phụ thuộc toàn cục, bao gồm những vấn đề:

- Trình bày một hệ thống các khái niệm và tiên đề trên các ràng buộc phụ thuộc cục bộ, phụ thuộc đường đi, phụ thuộc toàn cục, và phụ thuộc cố kết. Trên cơ sở đó, xây dựng các khái niệm bao đóng, phủ tối thiểu, thể hiện người sử dụng, thể hiện đối tượng, thể hiện tối đại; trình bày khái niệm mô hình đối tượng dựa trên khái niệm thể hiện tối đại. Đồng thời cũng nêu ra điều kiện cần và đủ của một mô hình đối tượng, như là một dấu hiệu để nhận biết một lớp đã được chuẩn hóa hay chưa;

- Đưa ra các qui tắc và thuật toán để chuẩn hóa đối tượng, dựa trên khái niệm thể hiện người sử dụng, bằng hai phương pháp: tái cấu trúc đối tượng và xây dựng lại đối tượng;

- Nêu ra các tiếp cận thiết kế mô hình dữ liệu hướng đối tượng, và đề xuất hai phương pháp sử dụng các nguyên tắc chuẩn hóa đối tượng được trình bày trong bài báo này để hỗ trợ thiết kế các lược đồ dữ liệu.

### TÀI LIỆU THAM KHẢO

[1]. C.Beerli, *A Formal Approach to Object – Oriented Databases, Data & Knowledge Engineering*, Vol.5, No.4, October 1990.

[2]. A.Dennis, B.H.Wixom, and D.Tegarden, *Systems analysis and design: An Object –Oriented Approach with UML*, John Wiley, Inc, 2002.

[3]. D.W.Embley, *Object database Development: Concepts and Principles*, Addition Wesley Longman, 1998.

[4]. A.Formica, H.D.Groger, *An Efficient Method for Checking Object-oriented Database Schema Correctness*, 23(3):333-368, 1998.

[5]. D.Maier, *Object – Oriented database Theory: An Introduction & Indexing in OODBS*, TU Muencen, haupt / WS2001, Database Hall of Fame, 2001.

[6]. C.S.R. Prabhu, *Object-Oriented Databas Systems: Approaches and Architectures*, Prentice Hall, 2001.

[7]. Z.Tari, J.Stokes, and S.Spaccapietra, *Object normal forms and dependency constraints for object-oriented schemata*, ACM Transactions on Database Systems, 22(4):513-569, 1997.

**ABSTRACT****AN APPROACH TO OBJECT-ORIENTED DATABASE  
NORMALIZATION**

In recent decades, the relational database model has made significant progress on databases. Along with rapid development of information technology, relational database has also encountered problems related to storing complex objects. Meanwhile, object-oriented database (OODB) could overcome these difficulties in processing and storing complex objects in the real world. But OODB has been meeting its own challenges, including a lack of the official norm accepted by ANSI or ISO like relational model; the theoretical basic of OODB has also continuously been researching and developing. Key problems in database such as normalization of OODB, theory of constraint and integrity are challenging scientists of information technology. The paper aims at researching an approach to normalization of OODB based on the concepts of path-dependent constraints in the object model suggested by Tari, Stokes and Spaccapietra [14].