

SỬ DỤNG METADATA ĐỂ CÀI ĐẶT CHI TIẾT CHO LỚP TRONG CƠ SỞ DỮ LIỆU HƯỚNG ĐỐI TƯỢNG

Nguyễn Minh Trang¹ và Phạm Thị Xuân Lộc²

¹ Khoa Kỹ thuật Công nghệ, Trường Cao đẳng Cần Thơ

² Khoa Công nghệ Thông tin & Truyền thông, Trường Đại học Cần Thơ

Thông tin chung:

Ngày nhận: 19/09/2015

Ngày chấp nhận: 10/10/2015

Title:

Using metadata to implement class details in object-oriented database

Từ khóa:

Metadata, cơ sở dữ liệu hướng đối tượng, tính thích nghi, lớp, thuộc tính, phương thức

Keywords:

Metadata, object-oriented database, adaptable, class, attribute, method

ABSTRACT

Nowadays, the object-oriented database management systems do not support design and execution in details to make them the most convenient. Concretely, the biggest challenge in transforming each class from the conceptual level to the new structure is the creation of its methods. This requires a lot of experiences and efforts because it has to be performed manually but the results are often difficult to reuse.

The article uses the metadata as an effective tool to remedy that defect, and illustrates it by a small software. There will be many choices for every attribute. Integrity constraints are verified appropriately to the context. All of this are always described, stored and updated automatically in the metadata.

From metadata, the software automatically forms various method kinds for any class. These kinds satisfy most user requirements for an information system. In each kind, the software introduces multiple signatures that users can select to match with class structure and the reality applying such database.

The solution is general so the idea and software presented here can be applied to the other platforms and contexts. Hence, the approach is effective in raising adaptation and reuse for designing and programming.

TÓM TẮT

Các hệ quản trị cơ sở dữ liệu (CSDL) hướng đối tượng hiện nay chưa hỗ trợ việc thiết kế và thực thi chi tiết sao cho tiện lợi nhất. Cụ thể là phải chuyển mỗi lớp từ mức quan niệm sang thành cấu trúc mới. Việc này đòi hỏi nhiều kinh nghiệm, công sức vì phải thực hiện thủ công nhưng kết quả khó tái sử dụng.

Bài viết dùng metadata như công cụ hữu hiệu để khắc phục khiếm khuyết đó và minh họa bằng một phần mềm nhỏ. Sẽ có nhiều chọn lựa cho thuộc tính được đưa ra. Các ràng buộc toàn vẹn được kiểm tra chặt chẽ. Tất cả đều được mô tả trong metadata, luôn được tự động lưu trữ và cập nhật.

Từ các metadata, phần mềm tự động hình thành đủ loại phương thức cho bất kỳ lớp nào. Các loại này được đề xuất khá phong phú, đáp ứng hầu hết các yêu cầu người dùng đối với một hệ thống thông tin. Ở mỗi loại, phần mềm giới thiệu nhiều định dạng (signature) để người dùng lựa chọn phù hợp với cấu trúc lớp và thực tiễn áp dụng CSDL đó.

Cách giải quyết là tổng quát nên ý tưởng và phần mềm có thể áp dụng cho các ngữ cảnh và platform khác. Cách tiếp cận mang lại hiệu quả không nhỏ trong việc nâng cao tính thích nghi, tính tái sử dụng của thiết kế và lập trình.

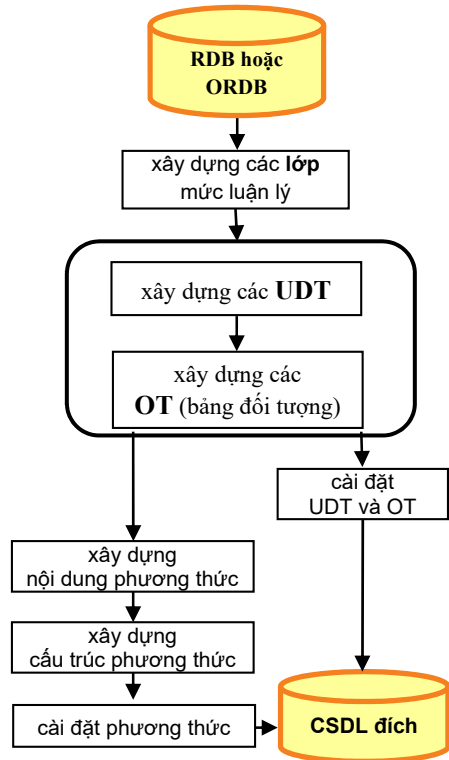
1 GIỚI THIỆU

Metadata (hay được dịch là siêu dữ liệu) là một thành phần dữ liệu dùng mô tả dữ liệu, cung cấp những thông tin cho phép người dùng hiểu rõ hơn bản chất về dữ liệu họ đang có.

Trên thế giới đã có một số nghiên cứu sử dụng metadata trong việc cài đặt cho các lớp như chuyển đổi CSDL quan hệ (RDB- Relational Database) sang các dạng CSDL hướng đối tượng (OODB- Object – Oriented Database), CSDL quan hệ - hướng đối tượng (ORDB- Object- Relational Database) và XML [3], chuyển đổi một RDB sang ORDB trong một hệ quản trị CSDL cụ thể là Oracle [4]. Quá trình chuyển đổi trong nghiên cứu [4] không có sự tương tác từ người dùng, chưa giải quyết trường hợp tham chiếu đệ quy, tham chiếu vòng, không có sự đa dạng khi xử lý các trường hợp tham chiếu giữa các bảng, cụ thể các thuộc tính tham chiếu chỉ được chuyển sang một dạng duy nhất là kiểu tham chiếu ref (reference) trong CSDL hướng đối tượng. Bài viết “Using the Metadata API” [11] cũng dùng metadata để xây dựng các phương thức cho phép người dùng truy xuất metadata của các đối tượng trong CSDL của họ. Tuy nhiên, các phương thức này được xây dựng trên các lớp có sẵn ở mức vật lý, trong khi chúng tôi muốn **xây dựng từ mức luận lý**, tức các lớp ở mức vật lý hoàn toàn chưa có.

Trong nghiên cứu của mình, chúng tôi khai thác triệt để metadata trong CSDL để phục vụ quá trình cài đặt chi tiết cho các lớp. Trước mắt, thực hiện cài đặt trên Oracle là một ORDB nhưng mục đích lâu dài của chúng tôi là đầu ra sẽ có trên các ORDB khác, cũng như trên các OODB thuần túy.

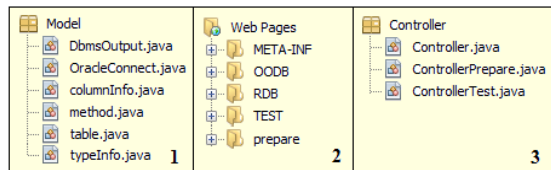
Bài viết tập trung vào việc hình thành tự động một số phương thức đủ loại cho bất kỳ lớp nào. Các phương thức đề xuất gồm phong phú các loại phương thức đáp ứng hầu hết các yêu cầu người dùng đối với một hệ thống thông tin. Phần cài đặt thuộc tính cũng được trình bày trong bài viết. Quá trình cài đặt có giải quyết vấn đề tham chiếu đệ quy, tham chiếu vòng, đưa ra nhiều tùy chọn cho từng công việc như cập nhật thuộc tính từ các lớp hiện có, hình thành các tham chiếu giữa các bảng (xem mục 4.3). Sơ đồ hệ thống được biểu diễn ở Hình 1.



Hình 1: Sơ đồ hệ thống

2 MÔI TRƯỜNG PHÁT TRIỂN

- Ngôn ngữ lập trình: Java.
- Hệ quản trị CSDL: Oracle 11gR2.
- Mẫu thực nghiệm: là các RDB và ORDB. Các mẫu thực nghiệm có đầy đủ các ràng buộc *primary key, unique, references, default, nullable, check*. Về tham chiếu, có đủ các dạng tham chiếu, các trường hợp tham chiếu vòng và tham chiếu đệ quy. Về kiểu dữ liệu, ngoài kiểu sơ cấp/cơ bản các mẫu còn có kiểu tập hợp (*collection*) như *nested table collections, varray collections*.
- Nền tảng xây dựng: ứng dụng web dựa trên mô hình MVC (Model-View-Controller). Hình 2 trình bày các thành phần trong mô hình MVC này.



Hình 2: Thành phần Model (1), thành phần View (2), Thành phần Controller (3) trong hệ thống

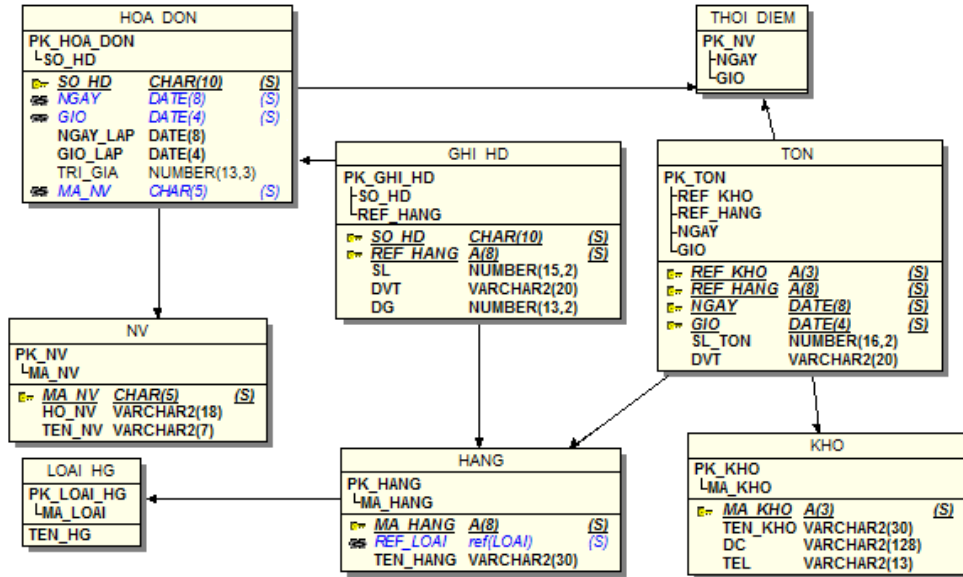
3 CÁC LOẠI PHƯƠNG THỨC

Các loại phương thức được đề xuất khá phong phú, đáp ứng hầu hết các yêu cầu người dùng đối với một hệ thống thông tin. Trong khuôn khổ bài báo này, do giới hạn về số trang nên các ví dụ được trình bày một cách ngắn gọn. Sơ đồ lớp của CSDL QuanLyHangHoa dùng trong các ví dụ được giới

thiệu ở Hình 3.

Các quy ước trong ví dụ:

- lh là một đối tượng đã có thuộc lớp LOAI_HG.
- h là một đối tượng đã có thuộc lớp HANG.
- k là một đối tượng đã có thuộc lớp KHO



Hình 3: Sơ đồ lớp ở mức luận lý của CSDL QuanLyHangHoa

Loại 1: Phương thức khởi tạo.

Vi dụ: taoLOAI_HG(1, 'thietbi'):LOAI_HG;

Loại 2: Phương thức hủy.

Vi dụ: lh.huyLOAI_HG();

Loại 3: Phương thức lấy ra trị của thuộc tính.

Vi dụ: lh.setTEN_HG('thuc pham');

Loại 4: Phương thức thay đổi trị thuộc tính.

Vi dụ: h.getREFLOAI(): ref(Loai);

Loại 5: Phương thức hiển thị đối tượng đang xét.

Vi dụ: h.hienHANG();

Loại 6: Phương thức hiển thị các đối tượng của lớp đang xét.

Vi dụ: hienDSHANG();

Loại 7: Như loại 6, nhưng có sắp theo nhóm.

Vi dụ: hienDSHANGtheoTEN_HANG();

Loại 8: tìm đối tượng theo khóa (hoặc theo OID) của lớp đang xét.

Vi dụ: timHANG (1): Hang;

timHANG('0E3C09BA16F6'): Hang;

Loại 9: Phương thức tìm các đối tượng của một lớp.

Vi dụ:

dsHANG (); HANG[]; dsTONtheoNGAY('1/1/15'): TON[]; dsTONtheoNGAY('1/1/15','1/5/15'):TON[];

Loại 10: Phương thức thống kê dùng hàm kết tập, chỉ tính trên toàn bộ lớp đang xét. Các ví dụ ở loại 10 này được xét trên lớp TON.

- *Nhóm 1:* tính tổng số đối tượng của một lớp.

Vi dụ: tsTON(): int;

- *Nhóm 2:* tính tổng số đối tượng của lớp theo tham số vào là thuộc tính lớp.

Vi dụ: tsTONtheoNGAY ('1/1/15'): int; tsTONtheoNGAY('1/1/15','1/5/15'):int;

- *Nhóm 3:* thực hiện các thống kê trên một thuộc tính của lớp.

Vi dụ: sumSL_TON(): int; maxSL_TONtheoNGAY('1/1/15'):int;

– **Nhóm 4:** tính tổng số đối tượng của một lớp dựa trên đối tượng (khóa của đối tượng) được tham chiếu đến.

Vi dụ: tsTONcuaHANG(‘H01’): int;

Loại 11: Phương thức tìm các đối tượng/ OID của ít nhất một lớp khác, có liên kết đến lớp đang xét.

Vi dụ: h.dsTON(): TON[];

h.dsTONtheoNGAY(‘1/1/15’): TON[];

Loại 12: như loại 11, nhưng có hiển thị kết quả sau khi tìm.

Vi dụ: h.hienDSHang ();

h.hienDSTONtheoNGAY(‘1/1/15’);

h.hienDSTONtheoNGAY(‘1/1/15’, ‘1/6/15’);

Loại 13: Phương thức kiểm tra một trị có là khóa của lớp đang xét không.

*Vi dụ:*kiemTraKhoa(‘100’): bit;

Loại 14: Phương thức tìm các đối tượng của lớp đang xét có một thuộc tính trong lớp đạt trị min/ max. *Vi dụ:*

HOA_DONcoTRI_GIAcaonhat:HOA_DON[]

Loại 15: như loại 14, nhưng xét dựa trên thuộc tính liên kết.

Vi dụ: KHOcoSL_TONlaMAX(): KHO[];

Loại 16: Phương thức tìm các đối tượng của lớp đang xét có một kết quả tổng theo nhóm trên thuộc tính đạt trị min/ max.

– **Nhóm 1:** tìm các đối tượng của lớp đang xét có một kết quả tổng theo nhóm trên thuộc tính đạt trị min/ max. *Vi dụ:*

LOAI_HGconhieuHANGnhat():LOAI_HG[]

– **Nhóm 2:** thực hiện tương tự nhóm 1 nhưng có một tham số vào là thuộc tính của lớp liên kết đến lớp đang xét. *Vi dụ:*

HANGnhiềuTONnhat(‘1/1/15’):HANG[];

Loại 17: Phương thức tính số lượng phần tử trong thuộc tính kiểu Collection.

Vi dụ: k.soluongTEL(): int;

Loại 18: Phương thức thống kê dùng hàm kết tập trên lớp khác có liên kết đến lớp đang xét. *Vi dụ:* k.tsTON(): Int ;

k.tsTONtheoNGAY(‘1/1/15’): Int;

k.tsTONtheoNGAY(‘1/1/15’, ‘1/6/15’): Int;

Mỗi loại phương thức đưa ra các tiêu chí khác nhau để lập nên danh sách phương thức thuộc loại phương thức đó. Các tiêu chí dựa trên kiểu dữ liệu của thuộc tính, các ràng buộc toàn vẹn, liên kết giữa các lớp.

4 QUY TRÌNH CÀI ĐẶT

4.1 Trích xuất thông tin

CSDL đầu vào có thể là RDB hoặc ORDB. Từ CSDL này, hệ thống tiến hành trích xuất các thông tin trong metadata. Các thông tin trích xuất bao gồm: danh sách các bảng, các thuộc tính trong bảng, chi tiết về kiểu dữ liệu của các thuộc tính, các ràng buộc toàn vẹn, các liên kết giữa các bảng. Minh họa về thông tin trích xuất được trình bày ở Bảng 1.

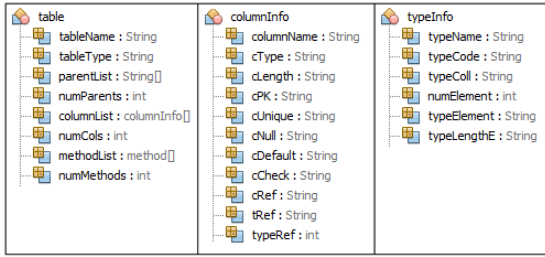
Bảng 1: Minh họa thông tin được trích xuất từ bảng SanPham

Bảng	thuộc tính	Kiểu	Kích thước	Khóa chính	Duy nhất	Rỗng	Trị mặc định	Điều kiện tham chiếu	Cột được tham chiếu	Bảng được tham chiếu	Kiểu tham chiếu
SanPham	MaSP	Number	4	P		N					
	TenSP	Varchar	20		U	N					
	XuatXu	Varchar	20			Y	Null				
	TrLg	Number	3			Y	0	> 0			
	MaLoai	Varchar	5			Y			MaLoai	LoaiSP	ref

4.2 Xây dựng DSR (Database Schema Representation)

DSR là một thành phần trong hệ thống, dùng lưu lại các dữ liệu được trích xuất từ CSDL nguồn.

DSR gồm ba mô hình chính là table (thông tin về bảng), columnInfo (thông tin về cột), typeInfo (thông tin về kiểu) (Hình 4). Các thông tin trong DSR một phần được trích xuất trực tiếp từ metadata, một phần cần qua phân tích và xử lý.



Hình 4: Mô hình table (1), mô hình columnInfo(2), mô hình typeInfo (3)

Ý nghĩa các thuộc tính được trình bày trong các Bảng 2- 4 như sau:

Bảng 2: Mô hình “table”

Thuộc tính	Tên bảng
<i>tableName</i>	tên bảng
<i>tableType</i>	kiểu của bảng
<i>listParent []</i>	danh sách tên các bảng được bảng hiện tại tham chiếu đến
<i>numParents</i>	số lượng bảng được tham chiếu đến
<i>listColumn []</i>	danh sách cột trong bảng
<i>numCols</i>	số lượng các cột
<i>listMethod</i>	danh sách phương thức trong bảng
<i>numMethods</i>	số lượng phương thức

Bảng 3: Mô hình “typeInfo”

Thuộc tính	Tên bảng
<i>typeName</i>	tên kiểu.
<i>typeCode</i>	loại kiểu (object hay collection).
<i>typeColl</i>	loại tập hợp (array hay table).
<i>numElement</i>	số phần tử trong kiểu tập hợp.
<i>typeElement</i>	kiểu từng phần tử trong tập hợp.
<i>typeLengthE</i>	chiều dài kiểu của phần tử trong tập hợp.

Bảng 4: Mô hình “columnInfo”

Thuộc tính	Tên bảng
<i>cType</i>	kiểu của cột
<i>cLength</i>	chiều dài kiểu dữ liệu
<i>cPK</i>	xác nhận có là khóa chính hay không
<i>cUnique</i>	xác nhận tính chất duy nhất
<i>cNull</i>	xác nhận có cho phép null không
<i>cDefault</i>	giá trị mặc định,
<i>cCheck</i>	ràng buộc miễn giá trị của cột
<i>cRef</i>	tên cột được cột hiện tại tham chiếu đến
<i>tRef</i>	tên bảng được cột hiện tại tham chiếu
<i>typeRef</i>	trường hợp tham chiếu

Ban đầu, các thông tin cần thiết cho quá trình cài đặt được lưu trữ riêng lẻ trên nhiều khung nhìn metadata trong CSDL nguồn. Việc xây dựng DSR giúp hệ thống chỉ cần truy xuất CSDL nguồn một lần, trong khi kết quả của DSR được sử dụng nhiều lần. Cách lưu trữ trong DSR cũng tạo sự thuận lợi

trong việc truy xuất thông tin cần thiết phục vụ các quá trình sau này trong hệ thống.

4.3 Tùy chỉnh về thuộc tính

Thông tin từ DSR được phép tùy chỉnh bởi người dùng. Các tùy chỉnh có thể thực hiện như thay đổi các thông tin của thuộc tính, thêm hoặc xóa thuộc tính, thay đổi tham chiếu giữa các bảng. Có ba dạng tham chiếu được đưa ra: tham chiếu obj (dạng lưu trữ như đối tượng), tham chiếu ref và khóa ngoại. Các thao tác cập nhật này được kiểm tra chặt chẽ trước khi thực hiện và có trình bày nội dung lỗi nếu thao tác thực hiện là không hợp lệ. Những thay đổi về dữ liệu sẽ được lưu trữ và cập nhật tự động trong DSR.

4.4 Cài đặt ORDB schema (lược đồ CSDL quan hệ - đối tượng)

Sau khi hoàn tất việc tùy chỉnh về thuộc tính, hệ thống tiến hành cài đặt ORDB schema. ORDB schema gồm hai thành phần chính là kiểu dữ liệu định nghĩa bởi người dùng (User defined type- UDT) và bảng đối tượng (Object table- OT). Thông tin của hai thành phần này được truy xuất và phân tích từ DSR.

4.4.1 Thông tin về UDT:

UDT sẽ thể hiện các thông tin của thuộc tính, các thông tin về các trường hợp tham chiếu ref. Quy tắc lưu thông tin trong UDT:

- Quy tắc 1: Mỗi bảng trong DSR sẽ chuyển thành một UDT.
- Quy tắc 2: Mỗi thuộc tính của bảng trong DSR sẽ trở thành thuộc tính của UDT tương ứng. Các thuộc tính sẽ được giữ nguyên về tên, kiểu dữ liệu và chiều dài kiểu.
- Quy tắc 3: Mỗi tham chiếu trong bảng DSR vẫn được bảo toàn và được thể hiện dưới một trong các dạng obj, ref, khóa ngoại.

Bảng 5: Minh họa bảng UDT

UDT	Thuộc tính	Kiểu	Kích Dạng thước.chiêu	UDT được t.chiêu
SanPham_Typ	MaSP	Number	5	
	TenSP	Varchar	20	
	XuatXu	Varchar	20	
	TrLuongNumber		5	
	MaLoai	Number	5ref	Loai_Typ

4.4.2 Thông tin về OT

OT biểu diễn thông tin về các ràng buộc của các thuộc tính. Quy tắc lưu thông tin trong OT:

– Quy tắc 1: Mỗi bảng trong DSR sẽ chuyển thành một OT.

– Quy tắc 2: Các ràng buộc trong DSR phải được thể hiện đầy đủ trong OT.

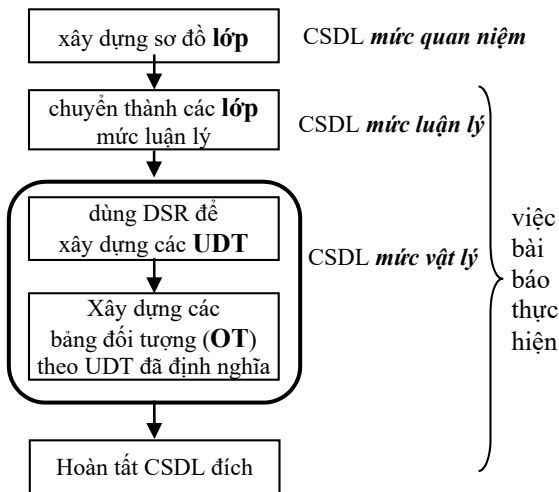
Bảng 6 minh họa thông tin trong OT.

Bảng 6: Minh họa cho OT

Đối tượng	Thuộc tính	Khóa	Duy nhất	Null	Trị mặc nhiên	Miền giá trị	Dạng Tham chiếu	OT được t.chiêu
Sanpham	MaSP	P						
	TenSP		U	N				
	XuatXu				Null			
	TrLuong				0	> 0		
	MaLoại						ref	Loại

4.4.3 Cài đặt ORDB schema

Quá trình cài đặt ORDB schema được trình bày tổng quát qua sơ đồ ở Hình 5.



Hình 5: Quá trình cài đặt lược đồ ORDB

a. *Xây dựng UDT*: Quá trình thực hiện qua hai bước con như sau:

– *Bước một*: khai báo cho từng UDT, bỏ qua các thuộc tính tham chiếu đối tượng và ref.

– *Bước hai*: thực hiện sau khi đã tạo xong tất cả các kiểu trong UDT. Bước này sẽ thêm các ràng buộc tham chiếu đối tượng và ref vào các thuộc tính trong UDT.

Việc tách quá trình khai báo UDT thành hai bước giúp hệ thống tránh được lỗi khi rơi vào trường hợp tham chiếu vòng, tham chiếu đệ quy.

b. *Xây dựng OT*: Tương tự như xây dựng UDT, việc xây dựng OT cũng được thực hiện qua hai bước:

– *Bước đầu*: cài đặt các OT theo UDT đã định nghĩa. Các tham chiếu ref và khóa ngoại sẽ được bỏ qua trong bước này.

– *Bước hai*: thực hiện sau khi đã cài đặt xong tất cả các OT. Tại bước này, các ràng buộc tham chiếu ref và khóa ngoại sẽ được thêm vào OT tương ứng.

4.5 Cài đặt phương thức

Mỗi loại phương thức có đặc điểm riêng. Tuy nhiên, các phương thức được cài đặt đều đảm bảo tuân thủ các quy tắc chung, như sau:

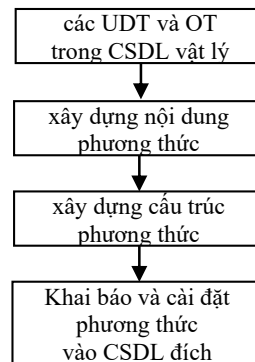
– Quy tắc 1: Mỗi phương thức phải được khai báo trước trong kiểu dữ liệu của bảng đối tượng tương ứng.

– Quy tắc 2: Mỗi phương thức đều có tên phân biệt. Tên mặc định của phương thức phải dễ nhớ.

– Quy tắc 3: Các phương thức phải cho kết quả chính xác.

– Quy tắc 4: Đối với các phương thức cập nhật dữ liệu, bên trong thân phương thức phải có phần kiểm tra ràng buộc toàn vẹn và thông báo lỗi tương ứng cho người dùng nếu lệnh gọi phương thức vi phạm các yêu cầu ràng buộc. Với các phương thức trình bày thông tin các đối tượng trong bảng đối tượng thì cần trình bày sao cho dễ đọc, thân thiện với người dùng.

Quá trình khai báo và cài đặt phương thức được trình bày tổng quát qua sơ đồ ở Hình 6.



Hình 6: Quá trình khai báo và cài đặt phương thức

4.5.1 Xây dựng nội dung phương thức

Từ dữ liệu trong DSR, hệ thống xác định danh sách các phương thức sẽ được sinh ra, trích và lưu lại các thông tin cần thiết cho việc khai báo và cài

đặt phương thức đó. Các thông tin được lưu thành từng bộ, mỗi bộ gồm các dữ liệu phục vụ xây dựng một phương thức. Tập hợp các bộ này được gọi là Method Content. Thông tin trong Method Content được minh họa ở Hình 7.

STT	LOAIPT	TENBANG	KIEUBANG	DANGPT	NHOMPT	TENCOOT	STTCOOT	KIEUCOOT	TOAMTU1	TOAMTU2	COTPHU1	COTPHU2	TENPT	TRANGTHAI
145	Static	NHANVIEN	NHANVIEN_TYP	9	5	NAMSIINH	2	NUMBER	min		GIOTINH	CHAR	minNAMSINHtheoGIOTINH	1
146	Static	NHANVIEN	NHANVIEN_TYP	9	5	NAMSIINH	2	NUMBER	sum		GIOTINH	CHAR	sumNAMSINHtheoGIOTINH	1
284	Static	PHONGBAN	PHONGBAN_TYP	15	2	ID_PB	2	REFPB	max	2	NHANVIEN	NAMSIINH	PHONGBANcoNhiềuNHANVIENNhato	1
285	Static	PHONGBAN	PHONGBAN_TYP	15	2	ID_PB	5	REFPB	max	2	NHANVIEN	GIOTINH	PHONGBANcoNhiềuNHANVIENNhato	1

Hình 7: Minh họa về các thông tin trong Method Content

4.5.2 Xây dựng cấu trúc phương thức

Method Statement là một mô hình được chúng tôi xây dựng trong hệ thống. Vai trò của mô hình này là sinh ra tự động cấu trúc (lệnh khai báo và cài đặt) cho các phương thức.

Mô hình Method Statement sẽ đưa ra cách xử lý cho mọi trường hợp có thể xảy ra trên mỗi loại phương thức. Chú trọng xử lý các trường hợp ràng buộc toàn vẹn, có cách xử lý riêng cho từng kiểu dữ liệu của thuộc tính như: kiểu căn bản/sơ cấp, kiểu tập hợp (array, table), kiểu tham chiếu (khóa ngoại, tham chiếu ref hay tham chiếu đối tượng), chú ý các ngoại lệ có thể có đối với từng dạng phương thức để đưa ra các thông báo lỗi đến người dùng khi họ gọi phương thức chưa chính xác.

Trong giai đoạn này, hệ thống sẽ truyền từng bộ thông tin trong Method Content vào mô hình

Method Statement. Mô hình Method Statement sẽ phân tích thông tin, sau đó trả về lệnh khai báo và lệnh cài đặt phương thức tương ứng.

4.5.3 Khai báo và cài đặt phương thức

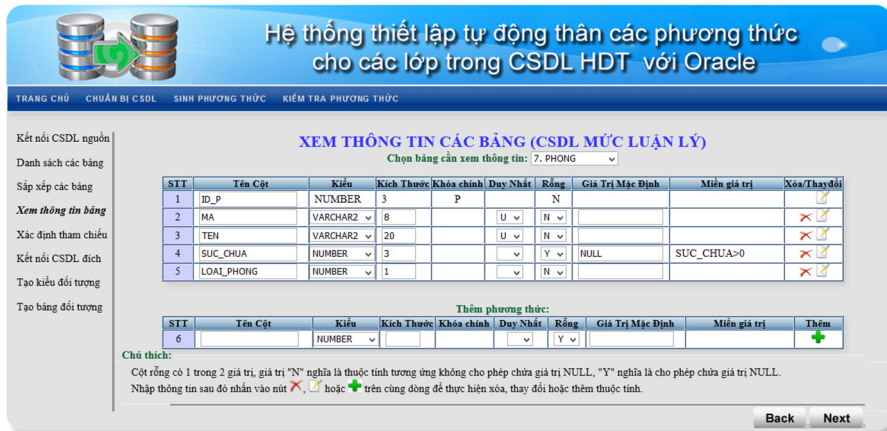
Từ kết quả xây dựng cấu trúc phương thức, hệ thống sử dụng lệnh khai báo để tiến hành khai báo cho các phương thức trong các UDT tương ứng và dùng lệnh cài đặt để cài đặt cho từng phương thức.

5 ĐIỂM MẠNH CỦA HỆ THỐNG

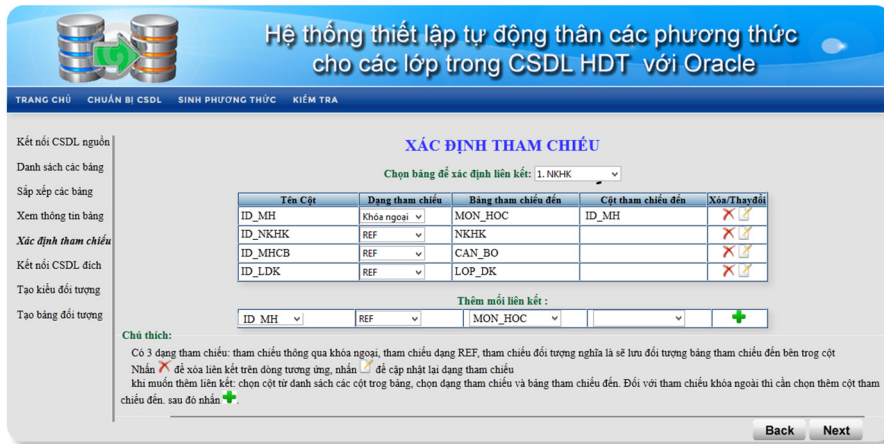
5.1 Tính linh hoạt, tùy biến trong xử lý

5.1.1 Tùy biến về thuộc tính

Trong quá trình cài đặt, hệ thống cho phép người dùng thực hiện xóa hoặc thêm thuộc tính, thay đổi thông tin thuộc tính (Hình 8), xác định lại tham chiếu giữa các bảng (Hình 9).



Hình 8: Giao diện các tùy chọn trên thuộc tính của bảng PHONG

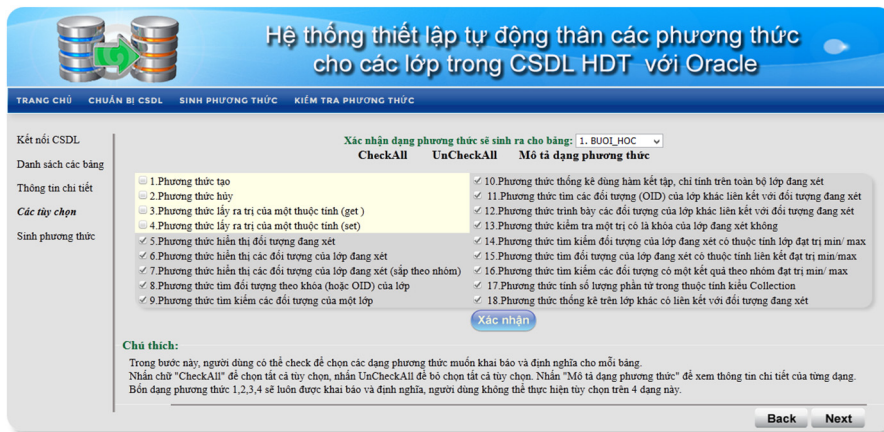


Hình 9: Giao diện các tùy chọn về tham chiếu

5.1.2 Tùy biến về phương thức

Ở mỗi loại phương thức, phần mềm giới thiệu nhiều định dạng (signature) để người dùng có thể lựa chọn phù hợp với cấu trúc lớp và với thực tiễn áp dụng CSDL đó. Chẳng hạn như người dùng có

thể chọn loại phương thức và từng phương thức cụ thể mà họ muốn hệ thống tiến hành sinh ra, đặt lại tên cho phương thức thay cho tên gợi ý, thay đổi toán tử so sánh trong các phương thức tìm kiếm. Hình 10 và 11 trình bày giao diện thực hiện các tùy chọn trên từng phương thức.



Hình 10: Giao diện thực hiện chọn loại phương thức sẽ cài đặt



Hình 11: Giao diện thực hiện tùy chỉnh trên từng phương thức

5.2 Tính độc lập trong xử lý

Quá trình cài đặt phương thức được tách thành hai phần đó là phần nội dung (Method Content) và phần lệnh (Method Statement) mang lại nhiều lợi điểm: tạo nên sự độc lập dữ liệu với quá trình xử lý, tăng tính linh hoạt, dễ dàng thay đổi tùy chọn cho các phương thức cần tạo..., giúp quá trình khai báo và định nghĩa trở nên đơn giản hơn.

5.3 Tính dễ bảo trì

Hệ thống được xây dựng theo mô hình MVC. Điều này làm cho mã lệnh trở nên trong sáng, dễ phát triển và dễ nâng cấp theo thời gian. Khi thay đổi giao diện bằng cách thay đổi thành phần View của nó thì Model (mô hình bên trong) cũng như Controller (bộ điều khiển) không hề thay đổi. Tương tự, ta có thể thay đổi Model hoặc Controller của đối tượng đồ họa mà những thành phần còn lại vẫn giữ nguyên.

5.4 Các chức năng hỗ trợ

Phần mềm minh họa có xây dựng một số chức năng phụ để hỗ trợ người dùng, tạo sự tiện lợi nhất khi sử dụng.

5.4.1 Cài đặt tự động CSDL mẫu

Hệ thống đưa ra sáu CSDL mẫu giúp người

dùng có thể kiểm nghiệm các chức năng của hệ thống ngay cả khi họ chưa có sẵn CSDL.

5.4.2 Chèn (Import), xuất (Export) lược đồ

Chức năng này giúp người dùng dễ dàng chèn lược đồ đã có vào hệ thống hoặc xuất kết quả cài đặt ra file .dmp.

5.4.3 Thực hiện kiểm tra tính đúng đắn cho các phương thức được phân mềm sinh ra

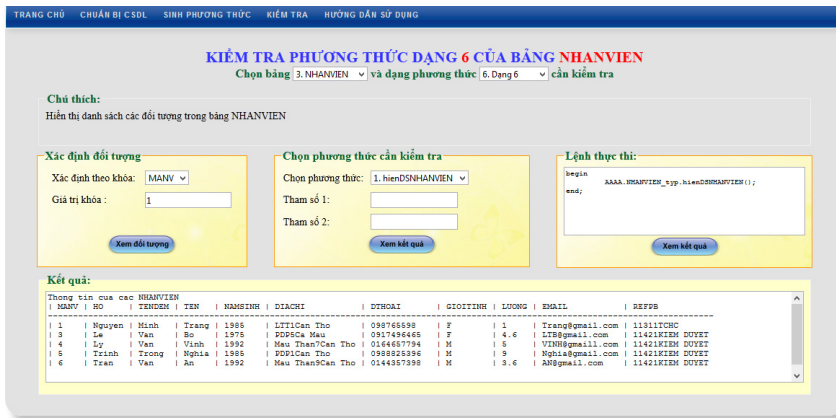
Hệ thống sử dụng thông tin trong Method Content để hỗ trợ cho quá trình kiểm tra tính đúng đắn của các phương thức. Giao diện thực hiện chức năng này được trình bày như Hình 12.

Giao diện gồm 3 phần chính

Phần chú thích: tạo mô tả và hướng dẫn một cách tự động theo mỗi phương thức được chọn.

Phần thực hiện phương thức: lọc ra các phương thức theo từng bảng, từng dạng phương thức và tự động trình bày kết quả khi người dùng nhấn chọn một phương thức.

Phần lệnh: Các lệnh dùng thực thi phương thức được chọn sẽ được trình bày trong phần lệnh. Phần này cho phép người dùng thay đổi lệnh để gọi phương thức theo ý của họ.



Hình 12: Giao diện chức năng kiểm tra

6 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Các tác giả đề xuất sử dụng metadata để cài đặt chi tiết cho lớp trong CSDL quan hệ đối tượng, CSDL hướng đối tượng và xây dựng thành công phần mềm minh họa cho ý tưởng đó. Phần mềm đơn giản nhưng đủ để chứa hết các trường hợp. Trong phần mềm sẽ có nhiều chọn lựa cho thuộc tính được đưa ra, cho phép thêm mối thuộc tính và kiểu thuộc tính từ các lớp hiện có, tự động hình thành một số phương thức đủ loại cho bất kỳ lớp nào. Các ràng buộc toàn vẹn cho thuộc tính được

kiểm tra chặt chẽ. Ngoài ra, nhằm tạo sự tiện lợi cho người sử dụng, phần mềm còn xây dựng một số chức năng phụ để hỗ trợ người dùng như chức năng cài đặt CSDL mẫu theo tùy chọn của người dùng, chức năng kiểm tra tính đúng đắn cho các phương thức được phân mềm sinh ra.

Ý tưởng đưa ra giúp giảm nhẹ thời gian, công sức cho những nhà thiết kế và lập trình trong việc khai báo và định nghĩa các phương thức đủ loại cho các lớp so với cách thực hiện thủ công. Cách tiếp cận mang lại hiệu quả không nhỏ trong việc

nâng cao tính thích nghi, tính tái sử dụng trong thiết kế và lập trình. Với cách giải quyết là tổng quát nên ý tưởng và phần mềm có thể áp dụng cho các ngữ cảnh và platform khác.

Phần mềm minh họa còn một số hạn chế, như: chưa hỗ trợ đa dạng các CSDL hướng đối tượng. Như vậy, trong thời gian tới, các tác giả cần khắc phục các hạn chế đã nêu, và thêm các tùy chọn cho người dùng như chọn ngôn ngữ lập trình hướng đối tượng, chọn cơ sở dữ liệu hướng đối tượng.

TÀI LIỆU THAM KHẢO

1. Đinh Khắc Quyền, bài giảng “Phân tích hệ thống”, Khoa Công nghệ Thông tin & Truyền thông, Trường Đại học Cần Thơ, 2000.
2. Phạm Thị Xuân Lộc, bài giảng “Phân tích thiết kế hướng đối tượng”, Khoa Công nghệ Thông tin & Truyền thông, Trường Đại học Cần Thơ, 2009.
3. Abdelsalam Amraga Maatuk, “Migrating Relational Databases Into Object Based And Xml Databases”, the School of Computing, Engineering and Information Sciences, Newcastle, 2009.
4. Phạm Thị Cao Ngân, luận văn Cao học “Xây dựng bộ công cụ chuyển đổi trong ORACLE một cơ sở dữ liệu quan hệ sang một cơ sở dữ liệu hướng đối tượng”, Khoa Công nghệ Thông tin & Truyền thông, Trường Đại học Cần Thơ, 2011.
5. Bulusu Lakshman, “Using Java Database Connectivity (JDBC) with oracle”, 2002.
6. Demix de Champeaux, Douglas Lea, Penelope Faure, “Object-Oriented System Development”, 1994.
7. Eric Belden, Janis Greenberg, Sundeep Abraham, Shashaanka Agrawal Geeta Arota, Yi Feng, Chandrasekharan Iyer, Geoff Lee, Bryn Llewellyn, Anand Manikutty, Valarie Moore, Magdi Morsi, Helen Yeh, Adiel Yoaz, Qin Yu. Oracle Database Object-Oriented Developer’s Guide, 11g Release 1 (11.1), Oracle Corporation, 2008.
8. Geeta Arota, Eric Belden, Chandrasekharan Iyer, Geoff Lee, Anand Manikutty, Valarie Moore, Magdi Morsi, Helen Yeh, Adiel Yoaz, Qin Yu. Oracle Database Application Developer’s Guide – Object-Relational Feature 10g Release 2 (10.1), Oracle Corporation, 2005.
9. Thomas M. Connolly, Carolyn E. Begg, Database systems: a practical approach to Design, Implementation and Management, (4E), Addison-Wesley, USA, 2005.
10. Lex de Haan, Sundeep Abraham, Drew Adams, Patrick, Oracle Database SQL Reference, 10g Release 2 (10.2), Oracle Corporation, 2005.
11. http://docs.oracle.com/cd/B19306_01/server.102/b14215/metadata_api.htm