TẠP CHÍ KHOA HỌC TÍNH TOÁN VÀ ĐIỀU KHIỂN TẬP V(1989), SỐ 1 (16-19)

OREST - Công cụ xây dựng hệ chuyên gia dựa trên biểu diễn đối tượng và luật.

Hồ Tú Bảo, Phạm Ngọc Khôi, Doãn Ngọc Liên, Đỗ Việt Nga.
Viện Khoa Học Tính Toán và Điều Khiển - Viện Khoa Học Việt Nam.

Trong bài này chúng tôi trình bày những nét chính về thiết kế và thực hiện hệ OREST, một hệ suy diễn tổng quát, cho phép ghép nối với các cơ sở tri thức biểu diễn đồng thời dưới dạng đối tượng và luật dẫn để tạo nên các hệ chuyên gia (HCG).

I. Những đặc điểm chính của OREST

1. Là một hệ lai (hybrid shell) cho phép đồng thời biểu diễn và sử dụng tri thức dưới dạng đối tượng (objects) và luật dẫn (production rules), thực hiện một số kiểu suy diễn khác nhau tùy theo yêu cầu của người sử dụng.
2. Là một hệ suy diễn mềm dẻo, linh hoạt cho phép dễ dàng thay đổi, thích nghi cho từng ứng dụng cụ thể.
3. Cho phép sử dụng hình ảnh giải thích cho tri thức. Các hình này được soạn thảo bằng các chương trình đồ họa như MS PAINT hoặc bằng Scanner.
4. Có khả năng học để tạo nên tri thức mới.
5. Có giao diện thân thiện với người sử dụng bằng các kỹ thuật của sổ, khả năng biểu diễn đồ thị của hệ tri thức, các giải thích hành động,...
6. Có khả năng lập luận với tri thức không chắc chắn, không chính xác theo logic không đơn điệu.

II. Các luận cứ khoa học của OREST

OREST có được nhiều tính chất mạnh của các hệ chuyên gia thế hệ thứ hai:

1/ Mội hạt nhân xây dựng hệ chuyên gia (Expert System Shell) đều gắn với một kiểu biểu diễn tri thức xác định nào đó. Kiểu biểu diễn tri thức (frames, rules, objects,...) là phương tiện cho người chuyên gia diễn đạt tri thức của mình. Và mỗi khi muốn đem dùng một nguồn tri thức trên máy tính, người ta phải nghĩ trước tiên đến một phương tiện thích hợp nhất để tải nguồn tri thức ấy.

Do mỗi kiểu biểu diễn tri thức đều có những hạn chế và do bản chất đa dạng, phức tạp của các hệ tri thức, việc tạo ra các công cụ mạnh (thoả mãn được tính đa dạng của tri thức) và dù mềm dẻo để biểu diễn và sử dụng tri thức là rất quan trọng. Đây là nguyên nhân ra đời của các hệ lai (hybrid), mà chủ yếu là "lai" về cách biểu diễn tri thức và cách suy diễn. Kiểu hệ lai được quan tâm rất nhiều hiện nay và trong thời gian tới đây là kiểu chấp nhận đồng thời biểu diễn tri thức bằng các đối tượng và luật dẫn, đồng thời thực hiện được suy diễn tiến và suy diễn lùi, suy diễn chính xác hay suy diễn xấp xỉ.

Lập trình hướng đối tượng, các ngôn ngữ lập trình đối tượng và các hệ chuyên gia dựa trên biểu diễn đối tượng đang rất được quan tâm trong trí tuệ nhân tạo. Trong cách biểu diễn này, người chuyên gia phải quan niệm và thể hiện tri thức của mình dưới dạng các đối tượng và xây dựng một cấu trúc phân cấp trên tập các đối tượng ban đầu này, bao gồm các lớp đối tượng và các quan hệ giữa chúng. Một lớp đối tượng (và bản thân chúng được xem là một đối tượng) là một tập các đối tượng có chung nhau một số tính cất nào đó. Các tính chất của mỗi đối tượng đều được truyền lại (kế thừa) cho các con cháu của nó.

Trong trường hợp đối tượng chỉ có một cha ta có sự kế thừa đơn. Một đối tượng cũng có thể có nhiều cha, khi này ta có sự kế thừa bội. Tri thức được biểu diễn trên một cấu trúc nhiều mức mà chỉ mức dưới cùng mới gồm các đối tượng ban đầu. Ưu điểm cơ bản của phương pháp này là cho ta thấy rõ các quan hệ phân cấp trong một lượng rất lớn tri thức (phân cấp theo các khái niệm tổng quát dần hoặc cụ thể hóa dần, theo tiến triển của thời gian, theo diễn biến của sự việc, theo quan hệ trước sau,...). Việc cấu trúc được tri thức như vậy cũng cho phép xây dựng một quá trình suy luận gọn hơn và hợp lý hơn.

OREST chấp nhận các hệ tri thức sử dụng đồng thời cách biểu diễn tri thức bằng luật dẫn và đối tượng. không có một sự phân biệt rõ ràng về việc dùng cách biểu diễn nào trong từng trường hợp cụ thể. Tuy nhiên về đại thể có thể phân biệt như sau:

Phần tri thức biểu diễn bằng các đối tượng về cơ bản là các tri thức tĩnh (static), có cấu trúc (có nhiều nhóm đối tượng mang tính chất chung), và thường mang tính mô tả...

Phần tri thức biểu diễn bằng luật dẫn về cơ bản là phần tri thức động (dynamic) thể hiện các mối quan hệ nhận quả (causal), các hành động phụ thuộc vào điều kiện là phần tri thức tản mạn, đơn lẻ, khó cấu trúc,...

Tóm lại một hệ tri thức trong OREST là một tập các đối tượng và các luật dẫn được viết theo cú pháp qui định của OREST và có một thứ tự tùy ý.

2/ Đại đa số những người làm nghiên cứu về hệ chuyên gia đã sớm đi đến kết luận: không có một cơ chế suy diễn vạn năng. Mỗi cơ chế suy diễn đều chỉ thực hiện được một số kiểu suy diễn nào đó: suy diễn tiến hay suy diễn lùi (hoặc cả hai), suy diễn với thông tin chính xác hay thông tin không chính xác,... và trong từng ứng dụng cụ thể cũng thường chỉ có một kiểu suy diễn nào đó là thích hợp nhất. Ý cơ bản của OREST là tạo ra một khối suy diễn độc lập tối đa với các phần khác, để có thể dễ dàng thay đổi khối này nhằm tạo ra nhiều chiến lược suy diễn khác nhau.

3/ Biểu diễn tri thức bằng hình ảnh được xem là một trong các đặc trưng đẹp nhất mà các công cụ xây dựng hệ chuyên gia ngày nay đang gắng đạt đến. Tri thức diễn đạt bằng các mệnh đề khi có thêm các hình giải thích cho phép người sử dụng dễ dàng trả lời chính xác các câu hỏi của hệ hoặc hiểu rõ hơn các kết luận hệ nêu ra.

4/ Khả năng được xem là các mục tiêu chủ yếu cần đạt được của các hệ chuyên gia trong thời gian tới đây. Thông qua việc học từ các thí dụ (examples), OREST sẽ tạo ra tri thức có thể sử dụng được trong quá trình suy diễn sau này, hoặc gọi giúp người kỹ sư tri thức tổ chức lại tri thức của mình. Hai hệ học của OREST là:

CABRO: Xuất phát từ tập các thí dụ quan sát của một số khái niệm tìm ra các luật suy diễn mô tả và nhận biết các khái niệm này.

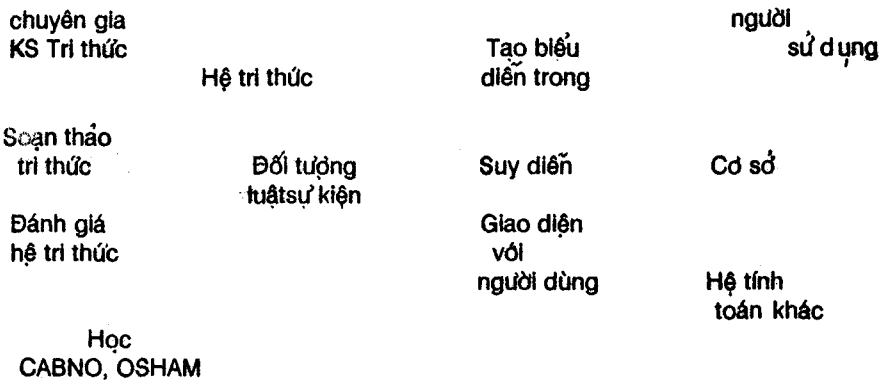
OSHAM: Xuất phát từ tập các đối tượng ban đầu, tạo tự động ra một cấu trúc phân cấp các đối tượng.

5/ Tính giao diện thân thiện (user - friendliness) là một trong các yêu cầu quan trọng hàng đầu của các sản phẩm tin học. OREST được xây dựng từ góc độ người sử dụng: dễ hiểu, dễ dùng nhờ các đối thoại tự nhiên, các lời giải thích, kỹ thuật cửa sổ, con chuột, màu sắc, âm thanh... cùng một cấu trúc chương trình hợp lý, khả năng dễ dàng theo dõi quá trình lập luận của hệ,...

6/ Nhờ cấu trúc đã nêu ở phần 2/, OREST cho phép ta dễ dàng thực hiện các kiểu lập luận với thông tin không chắc chắn và không chính xác, cho phép câu trả lời từ chối của người dùng ("I don't know") và cho những kết luận gần đúng.

III. Biểu diễn tri thức trong OREST

OREST bao gồm những khối chính sau đây:



Hình 1. Sơ đồ cấu trúc của hệ OREST

Ở dạng ngoài, cơ sở tri thức dùng được với OREST là một tệp chứa các đối tượng và các luật dẫn theo một thứ tự tùy ý.

CƠ SỞ TRI THỨC = {ĐỐI TƯỢNG} + {LUẬT}

Dạng chung của các đối tượng là:

OBJ: "Tên đối tượng"; /* lời giải thích */
SONOF: "Tên của đối tượng cha";
ATT: "Tên thuộc tính"; /* lời giải thích hoặc hình vẽ */
DOMAIN: Int ! Real ! Proposition ! String ! Object ;
DEFAULT: Tên của giá trị ngầm định;
WEIGHT: Số thực (trên đoạn [0,1] chỉ mức độ quan trọng của thuộc tính);
PREDICATES(.): Tập tân từ xác định tính chất của đối tượng;
METHODS: Biểu thức hoặc hàm gán trị cho một thuộc tính.
Các phương pháp định nghĩa trong một đối tượng sẽ được kế thừa cho toàn bộ các con cháu của nó;
ACTIONS: Tập các kết luận, các hành động.

Dạng chung của các luật là:

IF Predicates
THEN Actions

Phân tích kỹ hơn về cú pháp biểu diễn tri thức ta có:

OBJ: Tên đối tượng là một xâu kí tự bắt đầu bằng một chữ cái có độ dài tùy ý.
SONOF: Tên các cha trực tiếp của đối tượng đang xét.

Khai báo

SONOF cho phép đối tượng đang xét kế thừa các tính chất của ít nhất một đối tượng cha trực tiếp đã xác định trước.

ATTRIBUTES:Mỗi đối tượng non - terminal biểu diễn một lớp đối tượng có chung nhau một số thuộc tính nào đó. Các thuộc tính gồm các phần: tên thuộc tính, lời hoặc hình vẽ giải thích, miền xác định, trong số chỉ mức độ quan trọng của thuộc tính.

Tên của thuộc tính được xác định như tên của đối tượng; Sau phần tên thuộc tính, có thể viết thêm một đoạn giải thích cho thuộc tính này (hoặc một hình vẽ). Phần giải thích được viết giữa hai dấu /* và */.

Lưu ý thêm rằng các lời giải thích và các hình vẽ có thể được sử dụng một cách tương tự cho các đối tượng, các tân từ, các kết luận và hành động...

Có năm kiểu miềm xác định cho một thuộc tính được chấp nhận trong OREST: Integer, Real, Proposition, String và Object.

đối với các attributes kiểu In hoặc Real người dùng chỉ cần mô tả về kiểu:

Domain = Int;

hoặc Domain = Real;

Khi đó mỗi attributes có thể nhận mọi giá trị nguyên hoặc thực. Ngoài ra OREST còn cho phép định nghĩa khoảng xác định cho các thuộc tính kiểu nguyên và thực.

Domain = Interval of Int [10 ... 20];

Domain = Interval of Real [5. 3 .. 18.9];

Kiểu mệnh đề cho phép dùng các thuộc tính là một câu nói mang giá trị đúng hoặc sai. Người sử dụng có thể khai báo một mệnh đề ở dạng khẳng định hay phủ định (bằng cách thêm vào từ khóa "NOT" trước mệnh đề khẳng định).

Đối với các biến kiểu String có hai cách khai báo: Hoặc kê ra tất cả các giá trị có thể có:

Domain = (Xanh, Đỏ, Tím, Vàng);

hoặc ngầm định để OREST tự xác định các giá trị này trong khi đọc cơ sở tri thức. Khi này chỉ cần viết:

Domain = [ ];

Cuối cùng OREST cho phép miền xác định của một thuộc tính là một đối tượng khác:

Domain = OBJ : "Tên đối tượng";

Cần chú ý là khi này OREST không chấp nhận định nghĩa đệ quy, tức là một thuộc tính mô tả một đối tượng không được phép nhận chính đối tượng này làm miền xác định.

Giá trị ngầm định:

Mỗi thuộc tính được gắn một giá tri ngầm định. Khi không có sự định rõ của người sử dụng thuộc tính sẽ nhận giá trị ngầm định này trong quá trình suy diễn.

Trọng số: Đương nhiên mức độ quan trọng của mỗi thuộc tính (và cả của mỗi đối tượng), thậm chí của một thuộc tính trong các đối tượng khác nhau cũng có thể hoàn toàn khác nhau. OREST tự động quy ước mô tả mức độ quan trọng này bằng một số thực giữa 0 và 1.

Giá trị này càng lớn nói lên thuộc tính càng có tầm quan trọng và mỗi giá trị này sẽ được chuyên gia tri thức xác định trong toàn bộ mối tương quan chung giữa các thuộc tính. Sau khi đọc xong toàn bộ cơ sở tri thức, OREST tự động xác định một thang đo mức quan trọng trên toàn bộ các thuộc tính.

PREDICATES: Mỗi tân từ xác định một tính chất nào đó của đối tượng. Mỗi tân từ thường liên quan đến một hoặc nhiều thuộc tính được định nghĩa ở các đối tượng cha, hoặc là tân từ xác định các tính chất riêng của đối tượng (không liên quan đến các đối tượng khác)

Với các thuộc tính kiểu Int và Real, OREST cho phép sử dụng các toán tử so sánh sau đây:

"", "=", "", "=", "! ="

tương ứng chỉ lớn hơn, "lớn hơn hoặc bằng", "nhỏ hơn", "nhỏ hơn hoặc bằng", "bằng", "khác" và các hàm đại số thông dụng "abs", "sqrt", "sin", "cosin"...

Với các thuộc tính kiểu String, OREST cho phép sử dụng các toán tử so sánh, để chỉ sự bằng nhau và khác nhau giữa hai xâu ký tự.

Như vậy, mỗi predicate là một biểu thức logic của các biến nguyên, thực hoặc String.

METHOD: Mỗi đối tượng được gắn với một hay nhiều method(s) mô tả các thủ tục xác định giá trị cácthuộc tính nào đấy. Các thủ tục có thể là biểu thức sử dụng các toán tử so sánh, các toán tử quan hệ logic, các phép toán đại số trên các biến số và các biến String. Method có thể là tên một hàm kèm theo một danh sách biến.

ASTIONS: Nếu như các predicates xác định các tính chất ràng buộc của đối tượng và là phần cần phải đối sánh, thì các actions (còn có thể viết "coclusion") là phần mô tả các hành động cần phải thực hiện, các kết luận đạt được khi đối tượng được khẳng định đúng, hoặc khi toàn bộ các điều kiện (phần IF) của một luật được thỏa mãn.

Các actions có thể được mô tả bởi các mệnh đề. Dạng tổng quát của các hành động là các phép gán trị cho các biến số hoặc các biến String và có dạng:

A : = B (cho biến số)

A →B (cho biến String)

Trong đó B có thể là một hằng, một tên biến, một biểu thức hoặc một hàm số.

Kiểu của hàm số được xác định bởi ký hiệu [Int], [Real], [String] đặt trước tên hàm. Theo ngầm định hàm số có kiểu Real.

Đối với các luật, các điều kiện được xác định giống như các tân từ trong mô tả các đối tượng, và các hành động cũng giống như các hành động trong các đối tượng. OREST cho phép khi viết luật, người sử dụng có thể mô tả kiểu của các thuộc tính ở bất kỳ chỗ nào, trước hoặc sau một luật, hoặc ở một vùng chung cho tất cả các luật,... phân biệt bởi các từ khóa ATT.

IV. Lập luận của OREST

Hoạt động chính của OREST, sau khi đã nhận được nguồn giá trị thức và tạo thành biểu diễn trong, là trao đổi với môi trường bên ngoài để nhận thông tin về đối tượng cần xử lý, đối sách với nguồn tri thức và tạo ra các kết luận, các hành động cần thiết. Sơ đồ hoạt đông chính của OREST là:

START

START

Soạn thảo tri thức

Tạo biểu diễn trong

Khai báo các sự kiện

Xác định kết luận cần kiểm tra

Suy diễn để khẳng định hay phủ định kết luận

Suy diễn để rút ra các kết luận

STOP

Có thể sử dụng OREST để hoặc soạn thảo hệ tri thức hoặc suy diễn trên một cơ sở tri thức nào đó. OREST giúp người sử dụng soạn thảo tri thức một cách thông minh, có kiểm tra và phát hiện lỗi của tri thức. Sau khi soạn xong toàn bộ tri thức, OREST tự động dịch cơ sở tri thức thành dạng biểu diễn trong theo qui ước riêng của mình.

Khai báo các sự kiện liên quan đến đối tượng cần xử lý là quá trình OREST đối thoại với người sử dụng để thu nhận các thông tin ban đầu về đối tượng cần xử lý. Tương tự như vậy việc đối thoại này còn được tiến hành trong khi OREST lập luận.

Do đã tạo nên một cấu trúc phân cấp cho các đối tượng (mỗi đối tượng đều thuộc một mức nào đấy), Trình tự đặt câu hỏi để lấy thông tin là trình tự của các sự kiện xuất hiện theo thứ tự tăng dần của các mức. OREST sẽ liệt tất cả các thuộc tính xuất hiện ở một mức nào đó và đề nghị người sử dụng mô tả các thuộc tính. Sau khi chọn các thuộc tính, người sử dụng có thể xem danh sách các giá trị có thể nhận được của thuộc tính, đặc biệt cho các biến String, (hoặc interval cho các biến số) và chọn một giá trị nào đấy.

Mỗi khi một sự kiện được khẳng định, OREST sẽ bổ sung sự kiện này vào trong cơ sở sự kiện và truyền thông tin này trên toàn bộ cơ sở tri thức. Mỗi đối tượng hoặc luật sẽ ở một trong các trạng thái sau trong quá trình lập luận "nguyên", "mở", "đóng", "dây" tùy theo nó chưa được xét hoặc đã xét mà đang ở tình trạng đợi, không dùng được nữa hoặc các điều kiện xác định đã hoàn toàn thỏa mãn.

Tương tự như quá trình khai báo các sự kiện đầu, OREST đề nghị người sử dụng nếu các kết luận cần kiểm tra, cần phủ định hoặc khẳng định. Nếu người dùng nêu ra một kết luận OREST sẽ cố gắng vận dụng tri thức để trả lời theo nguyên tắc suy diễn lùi. Nếu không, OREST sẽ lập luận để rút ra tất cả các kết luận có thể có theo nguyên tắc suy diễn tiến.

Khi được hỏi về các sự kiện, người sử dụng có thể sẽ trả lời khẳng định phủ định hoặc từ chối (không biết). Khi có sự từ chối trả lời câu hỏi, OREST sẽ có một chiến lược xử lý riêng.

OREST cho phép người dùng trong quá trình suy diễn có thể xem các lời giải thích tri thức, xem các hình vẽ, xem cơ sở tri thức, cơ sở sự kiện, thêm bớt và sửa các sự kiện, xem mạng đối tượng.

Quá trình suy diễn của OREST là quá trình đối sánh giữa cơ sở tri thức và các sự kiện được khẳng định. Thứ tự đối sánh của các đối tượng và các luật trong OREST được xác định bằng một hàm chọn f định nghĩa như sau:

Giả sử trong phần phải đối sánh của một phần tử của tập tương có n thành phần, mỗi thành phần i có một trong số xác định trong tập [0, 1], với giá trị ngầm định là 1.

Hàm 1 được định nghĩa trên mỗi đối tượng:

Trong đó, P là số tân từ đã được khẳng định đúng. Nói đại để f xác định lượng thông tin, còn phải biết về phần tử đang đối sánh. thứ tự đối sánh trong tập các đối tượng, luật được xác định theo thứ tự tăng dần của hàm f.

V. Kết luận

OREST là công cụ xây dựng các hệ chuyên gia thế hệ thứ hai. OREST đạt được nhiều tính chất mạnh của các công cụ xây dựng hệ chuyên gia được bán nhiều nhất trên thị trường thế giới như KEE, S1, ART, NEXPERT - OBJECT... Nhiều nghiên cứu lý luận của các tác giả về bài toán học tự động, về biểu diễn tri thức đã được cải đặt trong hệ.

(Xem tiếp trang 29)