

Mô hình cơ sở tri thức về lập trình máy tính và thiết kế hệ truy vấn tri thức cho sinh viên ngành Công nghệ thông tin

Đỗ Văn Nhơn, Mai Trung Thành*
Trường Đại học Quốc tế Hồng Bàng

TÓM TẮT

Lập trình máy tính là một trong những kiến thức cốt lõi của sinh viên ngành Công nghệ thông tin (CNTT). Một ứng dụng hỗ trợ người học trong việc truy vấn trên tri thức lập trình, đáp ứng được độ tin cậy là rất cần thiết và có ý nghĩa trong việc cải thiện, nâng cao chất lượng đào tạo cho sinh viên ngành CNTT. Để thiết kế nhóm hệ thống này đặc biệt là các hệ thống đáp ứng độ tin cậy trong giáo dục, tri thức cần phải được biểu diễn, tổ chức lưu trữ trên máy tính làm cơ sở cho việc thiết kế và giải quyết các vấn đề truy vấn trên tri thức. Hiện nay có nhiều phương pháp cho việc biểu diễn tri thức về lập trình máy tính. Tuy nhiên, các phương pháp này còn gặp nhiều hạn chế, đặc biệt chưa hỗ trợ được nhiều dạng yêu cầu truy vấn từ người dùng. Bài báo này sẽ trình bày một mô hình tựa COKB cho thiết kế các lớp ứng dụng truy vấn trên cơ sở tri thức, bao gồm việc đưa ra mô hình biểu diễn tri thức lập trình máy tính, đồng thời đề xuất ngôn ngữ truy vấn trên mô hình tựa COKB cùng các kỹ thuật xử lý, thuật giải trong việc thực thi các yêu cầu được diễn đạt qua ngôn ngữ truy vấn. Kết quả nghiên cứu đã được vận dụng vào thiết kế hệ truy vấn tri thức hỗ trợ sinh viên học ngành CNTT.

Từ khóa: *Ontology COKB, biểu diễn tri thức và suy luận, hệ truy vấn tri thức, hệ cơ sở tri thức, hệ thống thông minh trong giáo dục*

1. ĐẶT VẤN ĐỀ

Thiết kế các lớp ứng dụng hỗ trợ học tập thông minh là rất cần thiết và ý nghĩa trong giáo dục [1, 2]. Kiến thức lập trình máy tính là một trong những kiến thức đặc biệt quan trọng đối với sinh viên ngành CNTT. Kiến thức lập trình máy tính cung cấp cho sinh viên các kiến thức cơ sở về lập trình như các khái niệm cơ bản, các quy ước, quy tắc trong lập trình để từ đó hình thành tư duy giải quyết các bài toán trong thực tế thông qua việc lập trình. Lập trình là nền tảng cơ bản bắt buộc để tiếp tục học các kiến thức khác trong khối ngành CNTT.

Một ứng dụng hỗ trợ truy vấn các kiến thức lập trình là rất cần thiết và ý nghĩa đối với người học, người dạy [1]. Truy vấn tri thức là một nhu cầu cơ bản của người học trong việc mong muốn tìm kiếm, hỏi hoặc truy xuất các thông tin có ý nghĩa trên cơ sở tri thức được xây dựng trước. Trên miền tri thức về lập trình máy tính nói chung, ta có thể phân thành một số loại yêu cầu truy vấn sau đây:

- **Yêu cầu tra cứu hay xác định một thuộc tính của yếu tố tri thức thuộc loại thành phần tri thức** như: Khái niệm, toán tử, hàm, bài toán, thuật giải/thuật toán. Ví dụ: Định nghĩa về kiểu dữ liệu, cho ví dụ về kiểu dữ liệu, chương trình, trình biên dịch, ký hiệu của phép toán cộng hai số nguyên, kiểu trả về và tính chất của phép toán nhân hai số

nguyên, tham số đầu vào của hàm strcpy, cách sử dụng hàm printf, định nghĩa hàmstrupr, phát biểu bài toán tìm ước chung lớn nhất của hai số nguyên dương.

- **Yêu cầu truy xuất hay liệt kê các tri thức**, ví dụ: Truy xuất các hàm hỗ trợ xử lý trên kiểu dữ liệu chuỗi, truy xuất thuật toán hỗ trợ giải bài toán sắp xếp trên mảng số nguyên.
- **Yêu cầu so sánh hai yếu tố tri thức**, ví dụ: So sánh lệnh for và while, so sánh hàmstrupr và strlwr, so sánh hàm strcpy và strncpy.
- **Yêu cầu cần xác thực hay kiểm chứng tính đúng/sai một thông tin về tri thức**, ví dụ: Thuật toán Binary Search là thuật toán hỗ trợ giải bài toán sắp xếp là đúng hai sai, độ phức tạp của thuật toán Selection Sort bằng với độ phức tạp của thuật toán Merge sort đúng hay sai.

Để hệ thống có thể đáp ứng các yêu cầu truy vấn này, cơ sở tri thức về lập trình trên máy tính cần phải được biểu diễn, tổ chức một cách phù hợp. Việc thiết kế, tổ chức một cơ sở tri thức phù hợp và hiệu quả sẽ là yếu tố quyết định để thiết kế các xử lý, cũng như bộ suy diễn giải quyết vấn đề một cách hiệu quả được đặt ra trên cơ sở tri thức [3]. Hiện nay, có nhiều nghiên cứu về các phương pháp biểu

Tác giả liên hệ: Mai Trung Thành

Email: thanhmt@hiu.vn

diễn tri thức cho thiết kế hệ thống truy vấn tri thức. Ta có một số kết quả về vấn đề này như sau:

Đồ thị tri thức (knowledge graph) là một công cụ hữu ích để biểu diễn nội dung cho web ngữ nghĩa [4], cũng như cho thiết kế cơ sở tri thức trong giáo dục. Trong công trình [5], tác giả trình bày các vấn đề truy vấn trên mô hình đồ thị tri thức. Kiến thức biểu diễn thông qua cách tiếp cận đồ thị, trong đó mỗi nút đại diện cho thông tin về một yếu tố tri thức, chẳng hạn như một khái niệm và mỗi cạnh đại diện cho mối quan hệ giữa các yếu tố tri thức. Cách tiếp cận này có hiệu quả đối với các ứng dụng tập trung vào việc biểu diễn các khái niệm với cấu trúc đơn giản và mối quan hệ giữa các khái niệm. Đối với kiến thức có cấu trúc khái niệm phức tạp hay phân loại đa dạng, chẳng hạn như kiến thức về toán tử, hàm, bài toán, thuật giải trong kiến thức về lập trình, các cách tiếp cận này chưa thật phù hợp.

Trong công trình [6] nhóm tác giả vận dụng mô hình tri thức quan hệ [7] cho thiết kế hệ hỗ trợ truy vấn trên miền tri thức về lập trình máy tính. Trong các công trình [8], các tác giả cũng trình bày một số giải pháp trong tích hợp các mô hình biểu diễn tri thức dựa trên sự kết hợp mô hình tri thức quan hệ và đồ thị tri thức, để thiết kế các hệ thống hỗ trợ yêu cầu truy vấn tri thức trên một số miền tri thức như toán cấp trung học phổ thông, lý thuyết đồ thị, cơ sở dữ liệu. Tuy nhiên, cách công trình này chưa quan tâm đến việc biểu diễn các thành phần tri thức khác như toán tử, hàm, các bài toán và các thuật toán. Và đặc biệt là chưa hỗ trợ được các ngôn ngữ truy vấn được đặc ra trên mô hình cơ sở tri thức. Vì vậy, các kết quả nghiên cứu này chưa đáp ứng được nhiều loại yêu cầu truy vấn cơ bản khác như chưa quan tâm đến các yêu cầu truy vấn về truy vấn tri thức có ràng buộc về điều kiện, cũng như các yêu cầu truy vấn về xác thực/kiểm chứng thông tin đặt ra trên cơ sở tri thức.

Dựa trên mô hình COKB [9], nhóm tác giả trong bài báo trong [10] đã trình bày phương pháp trong thiết kế hệ thống hỗ trợ truy vấn tri thức cho miền tri thức về Ngôn ngữ lập trình C/C++. Điểm mạnh của các hệ thống này là cung cấp các quy ước truy vấn có cấu trúc, kết quả của các hệ thống có độ tin cậy cao. Tuy nhiên, các kết quả nghiên cứu này chỉ đưa ra một vài cú pháp quy ước trong truy vấn, chưa xem xét một ngôn ngữ truy vấn tổng quát trên mô hình, cũng như chưa xem xét các yêu cầu truy vấn như sự so sánh và kiểm chứng thông tin.

Các ứng dụng được xây dựng dựa trên các mô hình

ngôn ngữ lớn (LLMs - Large Language Models) [11], như ChatGPT (Generative Pre-Trained Transformer), cũng như Copilot, Gemini. Điểm mạnh của các ứng dụng này là khả năng hỗ trợ rất đa dạng yêu cầu, đặc biệt cho phép giao tiếp bằng ngôn ngữ tự nhiên. Các ứng dụng này hiện nay còn tồn tại nhiều rủi ro đáng ngại, đặc biệt là trong lĩnh vực giáo dục và nghiên cứu [12]. Khi các hệ thống này đôi khi có thể cung cấp câu trả lời không chính xác và câu trả lời của chúng thường gặp phải ảo giác (Hallucinations) khi chúng cung cấp thông tin quá mức hoặc không liên quan. Một số vấn đề này cũng đã được chỉ ra trong công trình [12, 13]. Các hệ thống hỗ trợ đặc biệt lĩnh vực giáo dục cần phải đảm bảo tính tin cậy và phải phù hợp với các yếu tố như chương trình đào tạo, năng lực của người học.

Tương tự các hệ thống hỏi đáp (QAS - Question Answer Systems) dựa trên các kỹ thuật xử lý ngôn ngữ tự nhiên (NLP - Natural Language Processing) trong thiết kế các hệ thống trí tuệ nhân tạo (AI - Artificial Intelligence). Các nhóm hệ thống này cho phép thực hiện các yêu cầu truy vấn bằng ngôn ngữ tự nhiên và hệ thống sẽ trả về kết quả theo các yêu cầu. Có nhiều kỹ thuật giải quyết các vấn đề trong NLP như [14], trong đó cách tiếp cận dựa trên các kỹ thuật học sâu là một cách tiếp cận hiện đại trong thiết kế các loại hệ thống này QAS [15]. Tuy nhiên, hiện nay các giải pháp NLP dùng trong hệ thống AI, chẳng hạn như các hệ thống QAS chưa hỗ trợ thật sự tốt cho việc xử lý, giải quyết các vấn đề dựa trên cơ sở tri thức, đặc biệt là hỗ trợ được khả năng suy luận trên cơ sở tri thức.

Đối với các ứng dụng đòi hỏi độ tin cậy và chính xác về thông tin, như trong lĩnh vực giáo dục, các hệ thống dựa trên tri thức [16] được coi là một cách tiếp cận hiệu quả để thiết kế các hệ thống thông minh giải quyết các vấn đề được đặt ra trên cơ sở tri thức. Vì các cơ sở tri thức được xây dựng từ chuyên gia trong lĩnh vực này. Quá trình thiết kế và xây dựng cơ sở tri thức vô cùng chặt chẽ khi toàn bộ quy trình từ việc chọn nguồn, thực hiện thu thập, phân loại, kiểm tra, chuẩn hóa đều bắt buộc phải có sự tham gia, giám sát, cung cấp, góp ý đến từ các chuyên gia [17]. Nhìn chung, các nghiên cứu hiện tại như đề cập ở trên đều có những điểm mạnh riêng. Tuy nhiên, các mô hình thể hiện tri thức này cần được cải thiện để đáp ứng các yêu cầu cơ bản trong lĩnh vực giáo dục, chẳng hạn như sự phù hợp với người học, độ tin cậy và khả năng xử lý các yêu cầu truy vấn đa dạng từ người học.

Dựa trên khả năng biểu diễn được đa dạng thành phần tri thức của mô hình các đối tượng tính toán

COKB [9], bài báo này trình bày một mô hình tựa COKB (giản lược và bổ sung) hướng đến thiết kế hệ hỗ trợ truy vấn trên miền tri thức lập trình máy tính. Bằng cách đơn giản hay giản lược bớt độ phức tạp về cấu trúc của thành phần khái niệm cùng các bài toán liên quan. Bên cạnh đó, một số thành phần tri thức phổ biến trên miền tri thức lập trình sẽ được bổ sung gồm thành phần tri thức về bài toán và thuật giải. Từ mô hình cơ sở tri thức được đề xuất, một ngôn ngữ quy ước trong việc hỗ trợ diễn đạt các yêu cầu trong truy vấn tri thức cùng các xử lý cơ bản sẽ được đề xuất. Bên cạnh đó, các lớp bài toán truy vấn truy vấn tri thức và các thuật giải suy luận để xử lý và thực thi các yêu cầu truy vấn cũng sẽ được trình bày trong bài báo.

Kết quả nghiên cứu của bài báo được vận dụng vào cài đặt, thử nghiệm thành công hệ hỗ trợ truy vấn trên tri thức về lập trình máy tính (gồm các khái niệm, các kỹ thuật lập trình cơ bản, các cấu trúc dữ liệu, các thao tác cơ bản trên các cấu trúc dữ liệu và các thuật toán cơ bản trong lập trình). Hệ thống hỗ trợ hiệu quả cho các đối tượng là sinh viên ngành Công nghệ thông tin trong việc học kiến thức về lập trình máy tính.

2. PHƯƠNG PHÁP BIỂU DIỄN CƠ SỞ TRI THỨC VỀ LẬP TRÌNH MÁY TÍNH

2.1. Mô hình hóa tri thức về lập trình máy tính theo tiếp cận Ontology

Định nghĩa 2.1: Ta gọi một mô hình tựa COKB cho thiết kế các hệ truy vấn tri thức là một mô hình COKB-KQS gồm bộ 8 thành phần.

(C, H, R, Ops, Funcs, Problems, Methods, Rules)

C là tập các khái niệm trong miền tri thức, mỗi khái niệm $c \in C$ là một lớp các đối tượng. Mỗi đối tượng $o \in c$ được trang bị các đặc trưng gồm:

- Cấu trúc của một đối tượng gồm bộ (*Attrs*, *Props*, *InnerRules*). Trong đó, *Attrs* là tập các thuộc tính của đối tượng; *Props* là tập sự kiện để chỉ tính chất của đối tượng. *InnerRules* là tập các luật nội tại của một đối tượng. Trong đó mỗi luật r có dạng luật dẫn $h(r) \rightarrow g(r)$. Với $h(r)$ là tập sự kiện giả thiết và $g(r)$ là tập sự kiện kết luận của luật r .
- Các lớp bài toán gồm cùng các thuật giải: Tra cứu hay xác định giá trị của một thuộc tính; xác định tính chất của một đối tượng; xác định đối tượng; kiểm chứng giá trị của một thuộc tính bên trong đối tượng.

H là tập các quan hệ IS_A giữa các khái niệm trên tập C.

R là tập các quan hệ hai ngôi khác giữa các thành phần trong miền tri thức.

Ops là tập các toán tử trong miền tri thức.

Funcs là tập tri thức hàm trong miền tri thức.

Problems là tập các lớp bài toán trên miền tri thức, mỗi lớp bài toán p có cấu trúc gồm (*name*, *statement*, *input*, *output*, *examples*). Trong đó *name* là tên gọi của lớp bài toán p , *statement* là phát biểu bài toán của bài toán, *input* là các thông tin đầu vào của bài toán p , *output* là kết quả của bài toán p , *examples* là tập các ví dụ mẫu cụ thể của bài toán p .

Methods là tập các thuật giải. Mỗi m là một thuật giải cho bài toán p trong P. Thuật giải m có cấu trúc gồm (*name*, p , *idea*, *pseudocode*, *procedure*, *complex*, *prospects*, *consequences*, *visualizations*). Trong đó: *name* là tên của thuật toán m , $p \in P$ là bài toán với quan hệ là thuật giải m để giải bài toán p , *idea* là ý tưởng của thuật toán m , *pseudocode* là mã giả của thuật toán m , *procedure* là thủ tục của thuật toán m và được viết thành hàm f , *complex* là độ phức tạp của thuật toán m , *prospects* là những ưu điểm của thuật toán m , *consequences* là những hạn chế của thuật toán m , *visualizations* là tập các phần tử, mỗi phần tử là một minh họa tương ứng cho một bài tập cụ thể thuộc bài toán p .

Rules là thành phần tri thức luật trong miền tri thức. Mỗi luật có dạng luật dẫn $h(r) \rightarrow g(r)$. Với $h(r)$ là tập sự kiện giả thiết và $g(r)$ là tập sự kiện kết luận của luật r . Trên mô hình tri thức tựa COKB-KQS ta có xét 6 sự kiện theo định nghĩa dưới đây.

- **Sự kiện loại 1:** Sự kiện về khai báo tên của thành phần trong tri thức
- **Sự kiện loại 2:** Sự kiện về khai báo tên của một khái niệm/đối tượng, thuộc tính của một khái niệm/đối tượng; tên của một toán tử/thuộc tính của của một toán tử; tên của một hàm/thuộc tính phần của một hàm.
- **Sự kiện loại 3:** Sự kiện về sự xác định thuộc tính của một khái niệm/đối tượng/toán tử/hàm/bài toán, bằng một giá trị.
- **Sự kiện loại 4:** Sự kiện về sự bằng nhau giữa hai thuộc tính của khái niệm/đối tượng.
- **Sự kiện loại 5:** Sự kiện về sự phụ thuộc thuộc tính của một khái niệm/đối tượng vào các thuộc tính khác.
- **Sự kiện loại 6:** Sự kiện về mối quan hệ giữa khái niệm/đối tượng, hay giữa các thành phần khác trong tri thức.

Dựa trên các luật suy diễn, việc so khớp sự tương đồng hay hợp nhất giữa hai sự kiện là một trong những kỹ thuật cơ bản cho thiết kế các xử lý suy luận trên cơ sở tri thức. Ta có kỹ thuật so khớp sự

hợp nhất của hai sự kiện như sau:

Thuật giải 2.1 thuật toán so khớp hai sự kiện (*HopNhatSuKien*)

Input: f1, f2 Output: true/false	
1.	IF f1 $\equiv$ f2 THEN
2.	return true;
3.	END IF
4.	k1 $\leftarrow$ <i>LoaiSuKien</i> (f1);
5.	k2 $\leftarrow$ <i>LoaiSuKien</i> (f2);
6.	IF k1 $\neq$ k2 THEN
7.	return false;
8.	END IF
9.	IF k1 = 3 hoặc k1 = 4 hoặc k1 = 5 THEN
10.	IF <i>HopNhatSuKien</i> (RHS(f1), RHS(f2)) and <i>HopNhatSuKien</i> (LHS(f1), LHS(f2)) THEN
11.	return true;
12.	ELSE
13.	return false;
14.	END IF
15.	END IF
16.	IF k1 = 6 THEN
17.	IF f1[1] $\neq$ f2[1] THEN
18.	return false;
19.	END IF
20.	IF <i>HopNhatSuKien</i> (f1[2], f2[2]) and <i>HopNhatSuKien</i> (f1[3], f2[2]) THEN
21.	return true;
22.	END IF
23.	IF "SYMMETRY" $\in$ f1[4] and "SYMMETRY" $\in$ f2[4] THEN
24.	IF <i>HopNhatSuKien</i> (f1[2], f2[1]) and <i>HopNhatSuKien</i> (f1[3], f2[2]) THEN
25.	return true;
26.	END IF
27.	END IF
28.	END IF
29.	return false;

Hình 1. Trình bày thuật toán so khớp hai sự kiện

2.2. Ngôn ngữ đặc tả cho cơ sở tri thức COKB-KQS

Ngôn ngữ đặc tả mô hình hỗ trợ thiết kế hệ hỗ trợ truy vấn tri thức COKB-KQS được xây dựng dựa trên các quy ước đặc tả của mô hình tri thức COKB theo [9]. Ngôn ngữ đặc tả gồm:

- Tập các ký tự: Các ký tự bảng chữ cái (a, b, ..., z, A, B...Z), ký tự chỉ các phép toán (+, -, *, /, %), ký tự số (0-9).
- Tập các từ vựng: Từ (từ đơn, từ ghép), cụm từ, câu, đoạn, các tên định danh.
- Các kiểu dữ liệu: Các kiểu dữ liệu cơ bản, các kiểu dữ liệu có cấu trúc.
- Các biểu thức.
- Các lệnh và các cú pháp đặc tả cho từng thành phần trong tri thức.

3. NGÔN NGỮ TRUY VẤN VÀ BÀI TOÁN TRUY VẤN TRI THỨC

3.1. Ngôn ngữ truy vấn

Định nghĩa 3.1 Cho miền tri thức $\mathbb{K}$ được biểu diễn theo mô hình COKB-KQS (định nghĩa 2.1). Ta gọi

một ngôn ngữ truy vấn $\mathbb{L}$ trên $\mathbb{K}$ là một bộ gồm các thành phần sau đây.

$$\mathbb{L} = \langle \Sigma, Q, \Delta, S, P \rangle$$

- Σ là tập từ vựng, gồm các từ (từ đơn, từ ghép), cụm từ có nghĩa trên miền tri thức được đặc tả theo mô hình COKB-KQS;
- Q là tập các yêu cầu truy vấn, $Q = \{ \text{"tra cứu"}, \text{"liệt kê"}, \text{"so sánh"}, \text{"kiểm chứng"} \}$;
- Δ là tập các ngữ đoạn, ta có $\Delta = \{ \langle \text{câu-truy-vấn} \rangle, \langle \text{câu-dạng-1} \rangle, \langle \text{câu-dạng-2} \rangle, \langle \text{câu-dạng-3} \rangle, \langle \text{câu-dạng-4} \rangle, \langle \text{điều-kiện} \rangle, \langle \text{sự kiện-loại-1} \rangle, \langle \text{sự kiện-loại-2} \rangle, \langle \text{sự kiện-loại-3} \rangle, \langle \text{sự kiện-loại-4} \rangle, \langle \text{sự kiện-loại-5} \rangle, \langle \text{sự kiện-loại-6} \rangle, \langle \text{tên-quan-hệ} \rangle \}$;
- $S \in \Delta$ được gọi là tiên đề, trên miền tri thức lập trình máy tính ta có $S = \langle \text{câu-truy-vấn} \rangle$;
- P là tập các quy tắc sinh (production) có dạng $\alpha \rightarrow \beta$, trong đó $\alpha \in \Delta$, $\beta \in \Sigma \cup Q \cup \Delta \cup \gamma$, với γ là tập các quy tắc được định nghĩa khác. Trên miền tri thức lập trình máy tính, $\mathbb{K}$ ta có P gồm tập các quy tắc sau đây:

p1::=	<câu-truy-vấn> → <câu-dạng-1>
p2::=	<câu-truy-vấn> → <câu-dạng-2>
p3::=	<câu-truy-vấn> → <câu-dạng-3>
p4::=	<câu-truy-vấn> → <câu-dạng-4>
p5::=	<câu-dạng-1> → [{"tra-cứu", <sự-kiện-loại-2>}]
p6::=	<câu-dạng-2> → [{"liệt-kê", [<sự-kiện-loại-1>, <điều-kiện>]}]
p7::=	<câu-dạng-2> → [{"liệt-kê", [<sự-kiện-loại-1>, <điều-kiện>]}]
p8::=	<điều-kiện> → [{"<tên-quan-hệ>, <sự-kiện-loại-2>}, ...]
p9::=	<điều-kiện> → [{"<tên-quan-hệ>, <sự-kiện-loại-2>}, ...]
p10::=	<điều-kiện> → [{"<sự-kiện-loại-3>, <sự-kiện-loại-3>}, ...]
p11::=	<điều-kiện> → [{"<sự-kiện-loại-3>, <sự-kiện-loại-3>}, ...]
p12::=	<câu-dạng-3> → [{"so-sánh", [<sự-kiện-loại-2>, <sự-kiện-loại-2>]}]
p13::=	<câu-dạng-4> → [{"kiểm-chứng", <sự-kiện-loại-3>}]
p14::=	<câu-dạng-4> → [{"kiểm-chứng", <sự-kiện-loại-4>}]
p15::=	<câu-dạng-4> → [{"kiểm-chứng", <sự-kiện-loại-5>}]
p16::=	<câu-dạng-4> → [{"kiểm-chứng", <sự-kiện-loại-6>}]

Hình 2. Tập các quy tắc được định nghĩa khác

3.2. Bài toán truy vấn tri thức và thuật giải thực thi câu truy vấn

3.2.1. Phát biểu bài toán

Định nghĩa 3.2. Cho miền tri thức $\mathbf{K}$ được mô hình hóa theo COKB-KQS (định nghĩa 2.1) và ngôn ngữ truy vấn $\mathbf{Q}$ (theo định nghĩa 3.1). Cho trước tập các yêu cầu truy vấn $\mathbf{Qs}$, hãy xử lý và thực thi các yêu cầu truy vấn được đặc tả theo $\mathbf{Qs}$. Ta có thể mô hình hóa bài toán $\mathbf{P1}$ dưới dạng sau đây.

P1: $\mathbf{Qs} \rightarrow \mathbf{Kqs}$

$\mathbf{Qs} = \{S_1, S_2, \dots, S_n\}$, mỗi S_i là câu truy vấn tiên đề theo định nghĩa 3.1.

$\rightarrow$: là quá trình xử lý và thực thi các câu truy vấn trong S_i trong $\mathbf{Qs}$.

$\mathbf{Kqs}$ là tập các kết quả sau khi thực thi từng câu truy vấn S_i trong $\mathbf{Qs}$.

Ví dụ 3.1: Trên miền tri thức về kiến thức lập trình máy tính, người dùng mong muốn biết về “Chương trình là gì?”, hay “sự khác nhau giữa kiểu dữ liệu float và kiểu dữ liệu double”, hay “hàm getline() là hàm hỗ trợ thao tác nhập trên kiểu dữ liệu string có đúng không?”. Ta có thể mô hình hóa các yêu cầu trên thành các câu truy vấn theo định nghĩa 3.1 như sau:

$\mathbf{Qs} := \{["TRA_CUU", \text{CHUONG_TRINH.DINH_NGHIA}], ["KIEM_CHUNG", ["CAT_CHUOI", \text{substr}(), \text{string}]], ["SO_SANH", [\text{float}, \text{double}]]\}$. $\mathbf{Kqs}$ là tập các kết quả mà hệ thống xử lý, thực thi các yêu cầu truy vấn trong $\mathbf{Qs}$. Ta có thể sử dụng thuật giải tổng quát để tìm được $\mathbf{Kqs}$ bởi thuật giải được trình bày dưới đây.

3.2.2. Thuật giải tổng quát

Thuật giải thực thi câu truy vấn trong $\mathbf{Qs}$ để tìm $\mathbf{Kqs}$.

Thuật giải 3.1

Base: Cơ sở tri thức K (được mô hình hóa theo mô hình tri thức COKB-KQS định nghĩa 2.1)	
Input: $\mathbf{Qs}$; // Tập câu truy vấn $\mathbf{Qs} = \{S_1, S_2, \dots, S_n\}$, S_i theo định nghĩa 2.2.	
Output: $\mathbf{Kqs}$; // Kết quả thực thi các câu truy vấn trong $\mathbf{Qs}$	
1.	$\mathbf{Kqs} \leftarrow []$;
2.	FOR S in $\mathbf{Qs}$ do
3.	LoaiTruyVan = <i>KiemTraLoai</i> (S);
4.	IF LoaiTruyVan[2]=1 THEN
5.	“Xuất thông báo lỗi cú pháp”
6.	ELSE
7.	IF LoaiTruyVan[1] = 1 THEN
8.	$\mathbf{Kqs} \leftarrow \text{ThucThiCauTruyVan_dang_1}(S)$;
9.	END IF ;
10.	IF LoaiTruyVan[1] = 2 THEN
11.	$\mathbf{Kqs} \leftarrow \text{ThucThiCauTruyVan_dang_2}(S)$;
12.	END IF ;
13.	IF LoaiTruyVan[1] = 3 THEN
14.	$\mathbf{Kqs} \leftarrow \text{ThucThiCauTruyVan_dang_3}(S)$;
15.	END IF ;
16.	IF LoaiTruyVan[1] = 4 THEN
17.	$\mathbf{Kqs} \leftarrow \text{ThucThiCauTruyVan_dang_4}(S)$;
18.	END IF ;
19.	END IF
20.	END FOR
21.	return $\mathbf{Kqs}$;

Hình 3. Trình bày thuật toán tổng quát thực thi câu truy vấn

Thuật giải xử lý và thực thi *câu dạng 1*

<i>ThucThiCauTruyVan_Dang_1 (thuật giải xử lý và thực thi câu dạng 1)</i>	
1.	Bước 1: Khởi tạo các biến cần thiết cho chương trình obj, attr_obj;
2.	COMPS := C $\cup$ Ops $\cup$ Funcs $\cup$ Problems $\cup$ Methods
3.	Bước 2: Gọi obj là đối tượng trong S, attr_obj là thuộc tính cần xác định giá trị trong S;
4.	Bước 3: Thực hiện tìm các tri thức trong từng thành phần
5.	FOR comps in COMPS do
6.	FOR object in comps do
7.	IF HopNhatSuKien(object, obj) THEN
8.	FOR attr in ATTRIBUTES(object) do
9.	IF HopNhatSuKien(obj.attr_obj, LHS(object.attr)) THEN
10.	return attr;
11.	END IF
12.	END FOR
13.	END IF
14.	END FOR
15.	END FOR
16.	END FOR

Hình 4. Trình bày thuật toán thực thi câu truy vấn dạng 1

Thuật giải xử lý và thực thi *câu dạng 2*

<i>ThucThiCauTruyVan_Dang_2 (thuật giải xử lý và thực thi câu dạng 2)</i>	
1.	Bước 1: Khởi tạo các biến cho chương trình TP, TapDoiTuong, TapTam, toantu
2.	TapDoiTuong $\leftarrow \{\}$;
3.	TapTam $\leftarrow \{\}$;
4.	Bước 2: Gọi TP là loại thành phần tri thức trong câu truy vấn S;
5.	Bước 3: Nếu câu truy vấn S không có chứa điều kiện
6.	FOR tp in TP do
7.	TapDoiTuong $\leftarrow \{tp\}$;
8.	END DO
9.	return TapDoiTuong;
10.	Bước 4: Nếu S có chứa điều kiện
11.	IF KieuDuLieu(DIEUKIEN, 'set') THEN
12.	toantu := ' $\cap$ '
13.	ELSE
14.	toantu := ' $\cup$ '
15.	END IF
16.	Bước 5: Dò tìm các tri thức trong TP thỏa điều kiện
17.	TapTam $\leftarrow \{\}$;
18.	FOR dk in DIEUKIEN do
19.	IF LoaiSuKien(dk) = 4 THEN
20.	FOR obj in TP do
21.	FOR attr in ATTRIBUTES(obj) do
22.	IF HopNhatSuKien(dk, attr) THEN
23.	TapTam $\leftarrow \{obj\}$
24.	END IF
25.	END FOR
26.	END FOR
27.	ELSE
28.	IF KieuDuLieu(dk[1], 'string') and LoaiSuKien(dk[2]) = 2 THEN
29.	FOR obj in TP do
30.	FOR rela in R do
31.	IF HopNhatSuKien(dk[1], rela[1]) THEN
32.	IF HopNhatSuKien(rela[2], obj) and HopNhatSuKien(dk[2], rela[3])
33.	TapTam $\leftarrow \{obj\}$;
34.	ELSE IF "Symmetry" $\in$ rela[2] THEN
35.	IF HopNhatSuKien(rela[3], obj) and HopNhatSuKien(dk[2], rela[2])
36.	TapTam $\leftarrow \{obj\}$;
37.	END IF
38.	END IF
39.	END FOR
40.	END FOR
41.	END IF
42.	END IF
43.	IF TapDoiTuong = $\{\}$ THEN TapDoiTuong = TapTam;
44.	TapDoiTuong $\leftarrow$ TapDoiTuong toantu TapTam;
45.	END IF
46.	END FOR
47.	return TapDoiTuong;
48.	

Hình 5. Trình bày thuật toán thực thi câu truy vấn dạng 2

Thuật giải xử lý và thực thi *câu truy vấn dạng 3*

<i>ThucThiCauTruyVan_Dang_3 (thuật giải xử lý và thực thi câu dạng 3)</i>	
1.	Bước 1: Khởi tạo các biến trong chương trình, CacDG, CacDK
2.	CacDG ← [];
3.	CacDK ← [];
4.	ob1 = Lấy giá trị đối tượng thứ nhất trong S;
5.	ob2 = Lấy giá trị đối tượng thứ nhất trong S;
6.	Bước 2: Tìm điểm giống, khác giữa hai khái niệm/toán tử/hàm/bài toán/thuật giải
7.	FOR attr_ob1 in ATTRIBUTES(ob1) do
8.	FOR attr_ob2 in ATTRIBUTES (ob2) do
9.	IF <i>HopNhatSuKien</i> (attr_ob1, attr_ob2) THEN
10.	IF attr_ob1[2] = attr_ob2[2] THEN
11.	CacDG ← attr_ob1;
12.	END IF
13.	ELSE
14.	CacDK ← [attr_ob1, attr_ob2];
15.	END IF
16.	END FOR
17.	END FOR
18.	return [CacDG, CacDK];

Hình 6. Trình bày thuật toán thực thi câu truy vấn dạng 3

Thuật giải xử lý và thực thi *câu dạng 4*

<i>ThucThiCauTruyVan_Dang_4 (thuật giải xử lý và thực thi câu dạng 4)</i>	
1.	Bước 1: Đọc tri thức đưa vào các biến
2.	KBC ← Đọc tất cả tri thức về khái niệm;
3.	KBO ← Đọc tất cả tri thức toán tử;
4.	KBF ← Đọc tất cả tri thức về hàm;
5.	KBP ← Đọc tất cả tri thức bài toán;
6.	KBM ← Đọc tất cả tri thức về thuật giải;
7.	TriThucLuat ← Đọc tất cả tri thức luật;
8.	KBRelations ← Đọc tất cả tri thức về quan hệ;
9.	BaoDongTapSuKien ← KBC ∪ KBO ∪ KBF ∪ KBP ∪ KBM;
10.	Bước 2: SuKien ← Lấy sự kiện cần kiểm chứng trong S;
11.	Bước 3: Thực hiện suy diễn tìm tất cả sự kiện có thể sinh được từ tập luật
12.	flag ← true;
13.	WHILE flag do
14.	Tìm luật r trong TriThucLuat có thể áp dụng;
15.	IF r ≠ NULL THEN
16.	BaoDongTapSuKien ← r[2];
17.	ELSE
18.	flag ← false;
19.	END IF
20.	Áp dụng các quy tắc sinh sự kiện tự động
21.	+ Tự động sinh sự kiện loại 4 từ các sự kiện loại 3;
22.	+ Tự động sinh sự kiện loại 3 từ các sự kiện loại 4;
23.	+ Tự động sinh sự kiện loại 5 từ các sự kiện loại 3, loại 4
24.	IF Có sự kiện mới sinh ra THEN
25.	flag ← true;
26.	END IF
27.	END WHILE
28.	Bước 4: Kiểm tra sự kiện trong BaoDongTapSuKien
29.	FOR sk in BaoDongTapSuKien do
30.	IF <i>HopNhatSuKien</i> (sk[1], SuKien[1]) THEN
31.	return [true, sk];
32.	ELSE
33.	return [false, -1];
34.	END IF
35.	END FOR
36.	

Hình 7. Trình bày thuật toán thực thi câu truy vấn dạng 4

Ví dụ 3.2. Giả sử người dùng muốn truy vấn tìm “các thuật toán giải bài toán sắp xếp và có độ phức tạp tốt nhất là $O(n \cdot \log(n))$ ”. Ta có thể mô hình hóa các yêu cầu trên thành các câu truy vấn theo định

nghĩa 3.1 như sau: $Q_s = \{ [“LIET_KE”, METHODS, \{ [“GIAI”, BAI_TOAN_SAP_XEP], BestCaseComplex = O(n \cdot \log(n)) \}] \}$.

Áp dụng **thuật giải 3.1**, quy trình xử lý và thực hiện

câu truy vấn trong Qs như sau:

- *Giai đoạn 1:* Xác định loại truy vấn trong yêu cầu Qs, ta có yêu cầu trong Qs là *câu dạng 2*.

- *Giai đoạn 2:* Thực hiện gọi thuật giải *"ThucThiCauTruyVan_Dang_2"*. Quá trình thực thi của thuật toán này như sau:

BƯỚC	CÔNG VIỆC THỰC HIỆN	KẾT QUẢ HOẠT ĐỘNG CỦA THUẬT TOÁN
1	Khởi tạo biến	TapDoiTuong = {} TapTam = {}
2	Lấy điều kiện trong câu truy vấn S	DIEUKIEN = [{"GIAI", BAI_TOAN_SAP_XEP}, BestCaseComplex = O(n*log(n))]
3	Kiểm tra loại điều kiện	toantu = {}
4	Dò các tri thức thỏa điều kiện: ["GIAI", BAI_TOAN_SAP_XEP]	TapTam = {INTERCHANGE_SORT, SELECTION_SORT, INSERTION_SORT, BUBBLE_SORT, QUICK_SORT, HEAP_SORT, SHELL_SORT, MERGE_SORT, RADIX_SORT, COUNTING_SORT, BUCKET_SORT} TapDoiTuong = TapTam
5	Dò tri thức thỏa điều kiện có độ phức tạp tốt nhất là n*log(n). BestCaseComplex = O(n*log(n))	TapTam = {QUICK_SORT, HEAP_SORT, MERGE_SORT} TapDoiTuong = TapDoiTuong ∩ TapTam TapDoiTuong = {QUICK_SORT, HEAP_SORT, MERGE_SORT}

Hình 8. Minh họa quá trình hoạt động của thuật toán thực thi câu truy vấn dạng 2

4. THỬ NGHIỆM VÀ ĐÁNH GIÁ

4.1. Kiến trúc và các thành phần của hệ thống

Các thành phần của hệ truy vấn kiến thức Knowledge Querying Systems gồm:

(1) Thành phần giao diện là nơi cho phép người dùng (sinh viên, kỹ sư tri thức) có thể giao tiếp với hệ thống bằng cách nhập vào các yêu cầu, đồng thời cũng là nơi nhận các kết quả trả về từ hệ thống.

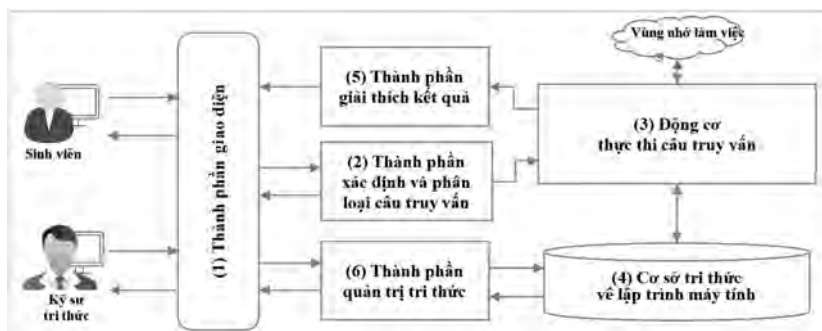
(2) Thành phần xác định loại và phân tích câu truy vấn đảm nhiệm hai vai trò. Thứ nhất là xác định các loại câu truy vấn cũng như lỗi nhập liệu. Thứ hai thực hiện phân tích cú pháp và gửi các kết quả phân tích đến động cơ thực thi câu truy vấn.

(3) Động cơ thực thi câu truy vấn là thành phần cho phép thực hiện các xử lý, quy tắc suy luận (được thiết kế theo thuật giải 3.1) để thực thi các yêu cầu truy vấn.

(4) Thành phần CSTT lập trình máy tính được biểu diễn theo mô hình COKB-KQS (định nghĩa 2.1).

(5) Thành phần giải thích kết quả cho phép thực hiện các xử lý các kết quả từ động cơ thực thi câu truy vấn giúp việc hiển thị các kết quả trở nên thân thiện với người dùng.

(6) Thành phần quản trị cơ sở tri thức là thành phần cho phép các kỹ sư tri thức thực hiện các thao tác trên tri thức như: Thêm, cập nhật, xóa các tri thức.



Hình 9. Kiến trúc của hệ truy vấn kiến thức Knowledge Querying Systems

4.2. Thiết kế cơ sở tri thức và động cơ thực thi câu truy vấn

Cơ sở tri thức được xem là thành phần quan trọng, quyết định về tính tin cậy của hệ truy vấn trên tri thức. Để đảm bảo được độ tin cậy và nhất quán trong toàn bộ cơ sở tri thức, việc thiết kế hay xây dựng cơ sở tri thức được thực hiện theo quy trình [17] gồm các giai đoạn: Thu thập tri thức, phân loại tri thức, chuẩn hóa cơ sở tri thức. Trong đó, có hai điều kiện bắt buộc để đảm bảo được chất lượng, sự tin cậy của cơ sở tri thức đó là:

- Nguồn tri thức thu thập cần phải đảm bảo độ tin cậy cao, gồm các tài liệu chuyên môn là các giáo trình, sách tham khảo, sách chuyên khảo, sách bài tập được sử dụng giảng dạy tại các trường đại học, xuất bản bởi các nhà xuất bản chính thống.

- Toàn bộ quá trình thu thập, phân loại, chuẩn hóa đều được giám sát, góp ý, cung cấp và đánh giá từ các chuyên gia, giảng viên có trình độ cao và kinh nghiệm trực tiếp giảng dạy các kiến thức lập trình.

Động cơ thực thi các truy vấn là một trong những thành phần cho phép xử lý, thực thi các yêu cầu truy vấn, thành phần này được thiết kế dựa trên thuật giải 3.1, bằng công cụ lập trình Maple.

4.3. Cài đặt thử nghiệm

Từ cơ sở động cơ truy vấn được thiết kế, đề tài cũng đã phát triển thành một ứng dụng web (dựa trên mã nguồn, C#, ASP.NET MVC), bên dưới đây là một phần giao diện của hệ thống.

Hệ thống đã thử nghiệm 200 yêu cầu truy vấn từ

sinh viên, giảng viên. Trong đó 169/200 (85%) có thể được diễn đạt qua các dạng câu truy vấn *câu dạng 1, câu dạng 2, câu dạng 3, câu dạng 4*. Bên dưới đây là bảng kê chi tiết về các loại yêu cầu.



Hình 10. Giao diện chỉ kết quả của việc thực thi các yêu cầu truy vấn theo ví dụ 3.1

Bảng 1. Bảng kê kết quả thực nghiệm hệ thống

STT	Loại yêu cầu truy vấn			Số lượng
	Dạng câu truy vấn	Yêu cầu	Chi tiết yêu cầu	
1	Câu dạng 1	Tra cứu thông tin của một yếu tố/thuộc tính của yếu tố tri thức thuộc loại tri thức	Khái niệm/thuộc tính của đối tượng	35
			Phép toán	9
			Hàm	25
			Lớp/dạng bài toán/thuộc tính của bài toán	8
			Thuật giải/Thuộc tính của thuật giải	8
2	Câu dạng 2	Liệt kê tri thức theo phân loại	Khái niệm, đối tượng, toán tử, hàm, lớp bài toán, thuật giải	5
		Liệt kê tri thức có ràng buộc điều kiện	Thuộc tính của một đối tượng/toán tử/hàm/bài toán/thuật giải bằng một giá trị.	24
			Thuộc tính của một đối tượng/toán tử/hàm/bài toán/thuật giải bằng thuộc tính của một khái niệm/toán tử/hàm/bài toán/thuật giải khác	7
3	Câu dạng 3	So sánh	Hai khái niệm	12
			Hai hàm	17
4	Câu dạng 4	Kiểm chứng một sự kiện	Thuộc tính của một đối tượng/toán tử/hàm bằng một giá trị	9
			Một thuộc tính của một đối tượng/toán tử/hàm bằng một thuộc tính của một khái niệm/toán tử/hàm khác	10
			Kiểm chứng một quan hệ giữa một thuộc tính của khái niệm/toán tử/hàm/bài toán/thuật giải với một thuộc tính của đối tượng/toán tử/hàm/bài toán/thuật giải	35
5	Chưa hỗ trợ	Nhận định về một yếu tố tri thức	Nhận định về một yếu tố tri thức khái niệm/hàm/thuật toán hoặc một thuộc tính của đối tượng/hàm/thuật toán	10
		Giải bài tập	Tạo một thuật giải tự động bằng một ngôn ngữ lập trình cụ thể	15
		Khác	Sự so sánh hai yếu tố tri thức dựa trên kết quả nhận định	6
	Tổng			200

4.4. So sánh và đánh giá

Kết quả ứng dụng của bài báo đã được so sánh với một số hệ thống cùng tính năng truy vấn trên miền tri thức lập trình máy tính. Trong đó bài báo xem

xét so sánh trên 3 tiêu chí gồm: *Tiêu chí sự tin cậy* (nghĩa là hệ thống cần đảm bảo được sự chính xác trong các thông tin hay kết quả được trả về); *Tiêu chí sự phù hợp* (nghĩa là hệ thống cần phải có sự hỗ

trợ cho nhóm người học cụ thể và phù hợp với chương trình đào tạo, đề cương chi tiết của học phần); *Tiêu chí về đáp ứng yêu cầu* (nghĩa là kết quả trả về của hệ thống phải phù hợp với yêu cầu từ người dùng, được diễn đạt qua câu truy vấn).

So với các hệ thống [6, 18], kết quả ứng dụng của bài báo chưa thể hỗ trợ tốt trong việc giao tiếp bằng ngôn ngữ tự nhiên. Tuy nhiên kết quả ứng dụng của bài báo thể hiện sự đa dạng yêu cầu hơn khi [6, 18] chỉ hỗ trợ được yêu cầu dạng 1 (*tra cứu thông tin*), dạng 3 (*so*

sánh hai khái niệm) và [10] chỉ hỗ trợ tra cứu dạng 1 (*tra cứu thông tin*) và dạng 2 (*liệt kê các tri thức theo phân loại*). Bên cạnh đó ChatGPT dù khá mạnh mẽ khi hỗ trợ được đa dạng yêu cầu và giúp giao tiếp bằng ngôn ngữ tự nhiên, tuy nhiên chúng lại có điểm yếu về sự tin cậy và kết quả thường gặp tình huống dư thừa thông tin so với yêu cầu. Đặc biệt là trong phạm vi mang tính đặc thù trong giáo dục, đòi hỏi một hệ thống thông minh cần trả lời đáp ứng độ tin cậy và phải đảm bảo được không dư thừa về thông tin.

Bảng 2. Bảng so sánh các ứng dụng liên quan

Các hệ thống	Hỗ trợ các yêu cầu				Sự tin cậy, phù hợp
	Câu dạng 1 - Tra cứu thông tin	Câu dạng 2 - Liệt kê tri thức/liệt kê tri thức có ràng buộc điều kiện	Câu dạng 3 - So sánh hai yếu tố tri thức khái niệm/hàm	Câu dạng 4 - Kiểm chứng sự kiện	
Hệ thống hỗ trợ học kiến thức Nhập môn lập trình [6]	Hệ thống chỉ hỗ trợ tra cứu thông tin khái niệm	Hệ thống không hỗ trợ truy vấn có ràng buộc điều kiện	Hệ thống chỉ hỗ trợ so sánh trên hai khái niệm	Hệ thống không hỗ trợ kiểm chứng sự kiện	- Hệ thống hướng đến là sinh viên đang theo học kiến thức nhập môn lập trình. - Kết quả trả về của hệ thống phù hợp với yêu cầu truy vấn, phù hợp việc dạy/học của giảng viên/sinh viên.
Hệ thống hỗ trợ học kiến thức Lập trình hướng đối tượng [18]	Hệ thống chỉ hỗ trợ tra cứu thông tin khái niệm	Hệ thống không hỗ trợ truy vấn có ràng buộc điều kiện	Hệ thống chỉ hỗ trợ so sánh trên hai khái niệm	Hệ thống không hỗ trợ kiểm chứng sự kiện	- Các nhóm hệ thống dựa trên cơ chế tiền xử lý ngôn ngữ vì vậy độ chính xác của hệ thống chưa được cao (61% TABLE I, trong [6]) và (79.22% TABLE IV, trong [18])
Hệ thống hỗ trợ học kiến thức về Ngôn ngữ lập trình C++ [10]	Hệ thống chỉ hỗ trợ tính năng tra cứu thông tin khái niệm/toán tử/hàm	Hệ thống có hỗ trợ tính năng truy vấn có ràng buộc điều kiện	Hệ thống không hỗ trợ tính năng truy vấn dạng so sánh hai khái niệm/hàm	Hệ thống không hỗ trợ tính năng kiểm chứng sự kiện	- Hệ thống hướng đến đối tượng là sinh viên đang theo học kiến thức lập trình hướng đối tượng. - Kết quả trả về của hệ thống phù hợp với yêu cầu truy vấn, phù hợp việc dạy/học của giảng viên/sinh viên. - Cơ sở tri thức được xây dựng từ các chuyên gia trực tiếp giảng dạy vì vậy kết quả trả về có độ tin cậy cao.
ChatGPT	Hệ thống có hỗ trợ tính năng tra cứu thông tin khái niệm/toán tử/hàm	Hệ thống có hỗ trợ tính năng truy vấn có ràng buộc điều kiện	Hệ thống có hỗ trợ tính năng truy vấn dạng so sánh hai khái niệm/hàm	Hệ thống hỗ trợ tính năng kiểm chứng sự kiện	- Kết quả của hệ thống cần có sự kiểm chứng lại thông tin vì các kết quả trả về của công cụ này có thể không chính xác. Vấn đề này cũng đã được trình bày trong công trình [13]. - Nội dung chưa phù hợp với chương trình đào tạo, đề cương chi tiết của học phần. - Cách trình bày của hệ thống cũng không thể hiện được trọng tâm của kết quả câu truy vấn.
Hệ truy vấn tri thức Lập trình máy tính	Hệ thống có hỗ trợ tính năng tra cứu thông tin khái niệm/toán tử/hàm	Hệ thống có hỗ trợ tính năng truy vấn có ràng buộc điều kiện	Hệ thống có hỗ trợ tính năng truy vấn dạng so sánh hai khái niệm/hàm	Hệ thống hỗ trợ tính năng kiểm chứng sự kiện (thuộc 6 loại sự kiện)	- Hệ thống hướng đến đối tượng người dùng là các sinh viên học lập trình về máy tính. - Kết quả trả về của hệ thống phù hợp với yêu cầu truy vấn, phù hợp việc dạy/học của giảng viên/sinh viên. - Cơ sở tri thức được xây dựng từ các chuyên gia trực tiếp giảng dạy vì vậy kết quả trả về có độ tin cậy cao.

So với các hệ thống [6, 18], kết quả ứng dụng của bài báo chưa thể hỗ trợ tốt trong việc giao tiếp bằng ngôn ngữ tự nhiên. Tuy nhiên kết quả ứng dụng của bài báo thể hiện sự đa dạng yêu cầu hơn khi [6, 18] chỉ hỗ trợ được yêu cầu dạng 1 (*tra cứu thông tin*), dạng 3 (*so sánh hai khái niệm*) và [10] chỉ hỗ trợ tra cứu dạng 1 (*tra cứu thông tin*) và dạng 2 (*liệt kê các tri thức theo phân loại*). Bên cạnh đó ChatGPT dù khá mạnh mẽ khi hỗ trợ được đa dạng yêu cầu và giúp giao tiếp bằng ngôn ngữ tự nhiên, tuy nhiên chúng lại có điểm yếu về sự tin cậy và kết quả thường gặp tình huống dư thừa thông tin so với yêu cầu. Đặc biệt là trong phạm vi mang tính đặc thù trong giáo dục, đòi hỏi một hệ thống thông minh cần trả lời đáp ứng độ tin cậy và phải đảm bảo được không dư thừa về thông tin.

5. KẾT LUẬN

Bài báo đã đưa ra được một mô hình cho việc biểu diễn được cơ sở tri thức về lập trình trên máy và bộ ngôn ngữ hỗ trợ truy vấn trên mô hình cùng các kỹ thuật xử lý cơ bản. Bộ ngôn ngữ này cho phép biểu diễn được một số yêu cầu phổ biến của người dùng. Trên cơ sở mô hình biểu diễn tri thức và bộ ngôn ngữ diễn đạt các yêu cầu truy vấn, bài toán truy vấn trên tri thức và các thuật toán để xử lý,

thực thi câu truy vấn cũng đã được trình bày trong bài báo. Kết quả nghiên cứu cũng đã vận dụng và cài đặt thử nghiệm thành công hệ hỗ trợ truy vấn tri thức dành cho các đối tượng là sinh viên ngành Công nghệ thông tin học về lập trình máy tính.

Bên cạnh các kết quả đạt được, kết quả nghiên cứu cũng còn một số điểm hạn chế như: Ngôn ngữ truy vấn được đề xuất trên mô hình còn chưa có sự đa dạng khi chỉ đáp ứng được 169/200 (85%) câu truy vấn được thu thập; Việc giao tiếp qua ngôn ngữ truy vấn chưa tự nhiên; v.v.

Các công việc tiếp theo của đề tài sẽ tập trung vào khắc phục các điểm yếu đã đề cập như tăng cường mở rộng các yêu cầu truy vấn chưa được xem xét xử lý trong bài báo. Tăng cường khả năng giao tiếp tự nhiên như xây dựng bộ chuyển đổi từ ngôn ngữ tự nhiên, sang ngôn ngữ truy vấn có cấu trúc quy ước. Đồng thời xem xét các vấn đề xây dựng tri thức dựa trên sự tích hợp, để từ đó làm cơ sở phát triển các lớp ứng dụng có khả năng hỗ trợ giải các yêu cầu truy vấn trên tri thức tích hợp.

LỜI CẢM ƠN

Nghiên cứu này được Trường Đại học Quốc tế Hồng Bàng cấp kinh phí thực hiện dưới mã số đề tài GVTC18.01.

TÀI LIỆU THAM KHẢO

- [1] Đ. V. Nhơn, "Các hệ thống trí tuệ nhân tạo ứng dụng trong giáo dục," *Hong Bang International University Journal of Science*, 2023.
- [2] S. A. Bilal Abu-Salih, "A systematic literature review of knowledge graph construction and application in education," *Heliyon journal*, vol. 10, no. 3, 2024.
- [3] Nhon V. Do, Vuong T. Pham Hien D. Nguyen, "A methodology for designing knowledge-based systems and applications," in *Applications of Computational Intelligence in Multi-Disciplinary Research*, 2022.
- [4] Roder, M., Kuchelev, D., Ngomo, AC.N, "A Topic Model for the Data Web," in *Knowledge Graphs and Semantic Web*, Lecture Notes in Computer Science, vol 14382. Springer, pp. 183-198, 2023.
- [5] A. Khan, "Knowledge Graphs Querying," *SIGMOD Record*, vol. 52, no. 2, pp. 18-29, 2023.
- [6] Xuan-Thien Pham, Tuan-Vi Tran, Van-Thanh Nguyen-Le, Vuong T. Pham, Hien D. Nguyen, "Build a search engine for the knowledge of the course about Introduction to Programming based on ontology Rela-model," in *International Conference on*

Knowledge and Systems Engineering, Can Tho, 2020.

- [7] Nhon V. Do, Hien D. Nguyen, Ali Selamat, "Knowledge-based model of expert systems using rela- model," *International Journal of Software Engineering and Knowledge Engineering*, vol. 28, no. 8, p. 1047 - 1090, 2018.

[8] Hung Nguyen, Thao T.N. Le, Hau Nguyen, Long D. Nguyen, Dung Dinh, Vuong T. Pham, Hien D. Nguyen, "Approach Design a Knowledge Chatbot System in Education Based on Ontology," in *International Conference on Intelligent Software Methodologies, Tools, and Techniques*, Mexico, 2024.

- [9] N. V. Do, "Ontology COKB for designing knowledge-based systems," in *Proceeding of 13th International Conference on Intelligent Software Methodologies, Tools, and Techniques*, 2014.

[10] T. T. M. Nhon V. Do, "knowledge representation model for designing the knowledge querying system in Programming Language C/C++," in *RIVF International Conference on Computing and Communication Technologies (RIVF)*, HaNoi, 2023.

- [11] M. A. K. Raiaan, M. S. H. Mukta, K. Fatema, N. M. Fahad, S. Sakib and M. M. J. Mim, "A Review on Large

Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges," *IEEE Access*, vol. 12, pp. 26839 - 26874, 2024.

[12] Mohammadreza Farrokhnia, Seyyed Kazem Banihashem, Omid Noroozi, Arjen Wals, "A SWOT analysis of ChatGPT: Implications for Educational Practice and Research," *Innovations in Education and Teaching International*, vol. 61, no. 3, pp. 460-474, 2024.

[13] C. B. J. L. F. D. V. H. F. S. P. K. K. U. Marek Urban, "ChatGPT can make mistakes. Check important info." Epistemic beliefs and metacognitive accuracy in students' integration of ChatGPT content into academic writing," *British Journal of Educational Technology*, 2025.

[14] Teotino Gomes Soares, Azhari Azhari, Nur Rokhman, Edi Winarko, "Proceedings of the International MultiConference of Engineers and

Computer Scientists," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, Hong Kong, 2021.

[15] X. Zhang, M. H. Chen and Y. Qin, "NLP-QA Framework Based on LSTM-RNN," in *International Conference on Data Science and Business Analytics (ICDSBA)*, 2018.

[16] P. S. Rajendra Akerkar, *Knowledge-Based Systems*, Jones & Bartlett Learning, 2010.

[17] Đ. V. Nhơn, *Các hệ cơ sở tri thức*, Đại học Quốc Gia TP HCM, 2021.

[18] Hien D. Nguyen, Tuan-Vi Tran, Xuan-Thien Pham, Anh T. Huynh, Vuong T. Pham, Diem Nguyen, "Design Intelligent Educational Chatbot for Information Retrieval based on Integrated Knowledge Bases," *International Journal of Computer Science*, vol. 49, no. 2, pp. 531-541, 2022.

Knowledge base model for computer programming and the design of knowledge query systems for information technology students

Do Van Nhon, Mai Trung Thanh

ABSTRACT

Computer programming is a core subject for students majoring in Information Technology (IT). An application that assists learners in querying knowledge about computer programming with high reliability is essential and meaningful in enhancing the quality of IT education. To design such systems, especially those that ensure reliability in educational contexts, knowledge must be represented and organized within computer systems to serve as a foundation for designing and addressing knowledge-based queries. Currently, various methods exist for representing programming knowledge. However, these methods still face numerous limitations, particularly in supporting diverse user query types and ensuring high reliability. This paper introduces a COKB based model for designing knowledge querying systems (KQS). It encompasses an ontology model for representing programming knowledge, a query language based on the proposed ontology model, the processing techniques and algorithms for executing requests expressed through this query language. The results have been applied in developing a knowledge query system to support IT students.

Keywords: Ontology COKB, knowledge representation and reasoning, knowledge-based systems, knowledge querying systems, intelligent software in education

Received: 07/5/2025

Revised: 12/6/2025

Accepted for publication: 14/6/2025