

DOI: <https://doi.org/10.59294/HIUJS.KHTT.2026.046>

PHÁT TRIỂN CÔNG CỤ QUẢN TRỊ CÁC CƠ SỞ TRI THỨC DẠNG COKB

Tô Hoài Việt*, Đỗ Văn Nhon, Mai Trung Thành, Huỳnh Vĩnh Phát
Trường Đại học Quốc tế Hồng Bàng

TÓM TẮT

Mô hình tri thức về các đối tượng tính toán COKB là một hướng tiếp cận hiện đại trong biểu diễn tri thức dựa trên ontology, cho phép mô hình hóa tri thức một cách có cấu trúc và hỗ trợ suy luận hiệu quả. Mô hình này đã được áp dụng thành công trong nhiều lĩnh vực như hệ chuyên gia, hệ hỗ trợ giải bài tập tự động thông minh và các hệ hỗ trợ truy vấn tri thức khác nhau. Tuy nhiên, việc triển khai và quản trị các cơ sở tri thức theo COKB vẫn còn gặp nhiều khó khăn do thiếu các công cụ hỗ trợ phù hợp. Các công cụ ontology hiện có chủ yếu dựa trên Description Logic, chỉ đáp ứng được các cấu trúc tri thức đơn giản và chưa hỗ trợ đầy đủ các thành phần như toán tử, hàm và luật suy diễn. Bài báo này đề xuất một hệ quản trị cơ sở tri thức được thiết kế chuyên biệt cho mô hình COKB. Hệ thống được xây dựng theo kiến trúc nhiều tầng, cung cấp các chức năng quản lý và truy vấn tri thức. Ngoài ra, bài báo trình bày chi tiết kiến trúc hệ thống, các thành phần chức năng và thuật toán xử lý truy vấn. Kết quả thử nghiệm cho thấy hệ thống hoạt động hiệu quả và khắc phục được những hạn chế của các công cụ ontology hiện nay.

Từ khóa: biểu diễn tri thức, hệ cơ sở tri thức, suy diễn, ontology

A MANAGEMENT TOOL FOR COKB-BASED KNOWLEDGE BASES

To Hoai Viet, Do Van Nhon, Mai Trung Thanh, Huynh Van Phat

ABSTRACT

The Computational Objects Knowledge Base (COKB) model is a modern ontology-based approach to knowledge representation, supporting structured knowledge modeling and efficient reasoning. This model has been successfully applied in various domains, such as expert systems, intelligent problem solving systems and many knowledge query systems. However, the implementation and management of knowledge bases based on COKB still meet several challenges due to the lack of appropriate supporting tools. Existing ontology tools are primarily based on Description Logic, which only supports simple knowledge structures and does not fully accommodate components such as operators, functions and inference rules. A Knowledge Base Management System (KBMS) specifically designed for the COKB model is proposed in this paper. The system is developed based on a multi-layer architecture, providing functionalities for knowledge management and querying. In addition, the paper presents the system architecture, functional components, and query processing algorithms in detail. Experimental results demonstrate that the proposed system operates effectively and addresses some limitations of existing ontology tools.

Keywords: knowledge representation, knowledge based system, automated reasoning, ontologies

1. ĐẶT VẤN ĐỀ

Mô hình tri thức về các đối tượng tính toán (COKB) [1] là một trong những mô hình biểu diễn tri

* Tác giả liên hệ: Tô Hoài Việt, Email: vietth@hiu.vn

(Ngày nhận bài: 06/3/2026; Ngày nhận bản sửa: 17/3/2026; Ngày duyệt đăng: 23/3/2026)

thực hiện đại theo tiếp cận Ontology, được trang bị một cơ sở lý thuyết hoàn thiện. Đã có nhiều kết quả nghiên cứu liên quan, và chứng minh được tính hiệu quả của chúng trong thiết kế các lớp ứng dụng trong thực tiễn [2]. Đặc biệt là các hệ chuyên gia, các lớp ứng dụng hỗ trợ học tập trong giáo dục [3, 4], cũng như nhiều lĩnh vực khác. Ta có thể kể đến một số hệ thống được thiết kế dựa trên nền tảng mô hình tri thức COKB như: hệ hỗ trợ giải bài tập tự động thông minh [5, 6], hệ hỗ trợ truy vấn tri thức, minh họa thuật giải và đánh giá kiến thức [7, 8], hay các hệ hỗ trợ chẩn đoán bệnh trong y học [9], hệ hỗ trợ tư vấn kiến trúc nhà ở. Tuy nhiên, việc triển khai và quản trị các cơ sở tri thức dạng COKB hiện nay gặp nhiều khó khăn do thiếu các công cụ hỗ trợ phù hợp. Các công cụ ontology phổ biến như Protégé chủ yếu dựa trên Description Logic [10, 11], chỉ hỗ trợ các cấu trúc tri thức đơn giản và chưa đáp ứng được các yêu cầu biểu diễn tri thức đa dạng như toán tử, hàm và luật suy diễn.

Vì vậy, việc phát triển công cụ hỗ trợ cho mô hình COKB sẽ góp phần rất ý nghĩa trong việc làm giảm các chi phí thiết kế, xây dựng và cài đặt các hệ cơ sở tri thức nói riêng, các hệ thống trí tuệ nhân tạo nói chung.

2. TỔNG QUAN NGHIÊN CỨU

Nghiên cứu được thực hiện dựa trên sự kết hợp giữa phương pháp biểu diễn tri thức theo ontology COKB và các công cụ quản trị ontology phổ biến hiện nay. Mô hình các đối tượng tính toán (COKB - Computational Object Knowledge Base) [1] gồm có 6 thành phần:

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

Trong đó, C là một tập hợp hệ thống khái niệm trong miền tri thức và mỗi khái niệm là một lớp đối tượng tính toán. Mỗi lớp các đối tượng tính toán được phân lập dựa trên cấu trúc hay cách xác định (định nghĩa) của khái niệm trong miền tri thức như sau:

- Tập các khái niệm cơ sở cấp 0: $C_{(0)}$: $C_{(0)} = \{c \mid \text{DOM}(c) \subseteq \mathbb{R} \text{ or } c \text{ là khái niệm tiền đề trong miền tri thức}\}$

- Tập các khái niệm cấp k: $C_{(k)} = \{c \mid (\forall x \in \text{Attrs}(c), \exists i < k, \exists c' \in C_{(i)}: x \in c') \wedge (\exists x \in \text{Attrs}(c), \exists c' \in C_{(k-1)}: x \in c')\}$, $k > 1$.

H là tập các quan hệ phân cấp giữa các đối tượng tính toán; R là tập các khái niệm về loại quan hệ giữa các đối tượng tính toán; Ops là tập các toán tử; Funcs là tập các hàm; Rules là tập các luật dẫn, trên mô hình COKB các sự kiện được xem xét theo 12 loại đặc trưng.

Protégé [12] là một công cụ được phát triển bởi Đại học Stanford (Stanford Center for Biomedical Informatics Research). Dựa trên nền tảng Logic mô tả (DL - Description Logic), Protégé cho phép xây dựng các cơ sở tri thức có thành phần và cấu trúc đơn giản, bao gồm các thành phần tri thức như *khái niệm, quan hệ, các cá thể, các tiền đề*. Các cơ sở tri thức được xây dựng từ Protégé được lưu trữ dưới dạng các tệp tin có cấu trúc chuẩn chuẩn Semantic Web như OWL (*.owl), RDF/RDFS (*.rdf), XML (*.xml), đồng thời cung cấp API lập trình và hệ thống plugin mở rộng, giúp người quản trị có thể tích hợp công cụ này vào các nền tảng tri thức tích hợp, v.v

Ngoài công cụ Protégé thì cũng có một số công cụ khác hỗ trợ việc quản trị ontology. Có thể kể đến như TopBraid Composer [13], GraphDB [14], VocBench [15]. Các công cụ này đều hỗ trợ quản trị ontology, tuy nhiên cũng tương tự như Protégé, dựa trên nền tảng là Logic mô tả, vì vậy các công cụ này cũng chỉ dừng lại cho việc xây dựng các cơ sở tri thức có cấu trúc đơn giản và hạn chế loại thành phần. Vì vậy, việc thiết kế các cơ sở tri thức đa dạng thành phần, có cấu trúc khái niệm trừu tượng và phức tạp chỉ các công cụ này sẽ gặp phải giới hạn của chúng.

Bài báo này đề xuất một hệ quản trị cơ sở tri thức (KBMS - Knowledge Base Management System) được thiết kế dành riêng cho mô hình COKB. Hệ thống cung cấp các chức năng quản trị và truy vấn tri thức được biểu diễn dựa trên mô hình COKB, với kiến trúc nhiều tầng tương tự các hệ quản trị cơ sở dữ liệu hiện đại. Bài báo trình bày chi tiết kiến trúc hệ thống, các thành phần chức năng, ngôn ngữ truy vấn và thuật toán xử lý truy vấn. Kết quả thử nghiệm cho thấy hệ thống có khả năng quản lý và

suy luận tri thức hiệu quả, đồng thời khắc phục được các hạn chế của các công cụ ontology hiện có.

3. PHƯƠNG PHÁP NGHIÊN CỨU

Nghiên cứu này tập trung đề xuất và xây dựng hệ quản trị cơ sở tri thức dựa trên nền tảng ứng dụng của các hệ cơ sở tri thức được xây dựng theo dạng mô hình ontology COKB. Trên cơ sở phân tích các mô hình biểu diễn tri thức, kiến trúc của các ứng dụng đã xây dựng và các cơ chế xử lý truy vấn, một kiến trúc KBMS phù hợp với đặc thù các tri thức được biểu diễn theo dạng ontology COKB được xây dựng.

3.1. Giới thiệu về kiến trúc KBMS

KBMS (Knowledge Base Management System) là hệ quản trị cơ sở tri thức được thiết kế nhằm quản lý, lưu trữ và hỗ trợ suy luận trên mô hình tri thức COKB (Computational Object Knowledge Base). Khác với các hệ quản trị cơ sở dữ liệu truyền thống (DBMS) như MySQL hay PostgreSQL - vốn tập trung quản lý dữ liệu dạng bảng (table), KBMS được xây dựng để quản lý các thành phần tri thức có cấu trúc phức tạp, bao gồm:

- **Concepts (C):** Tập các khái niệm
- **Hierarchy (H):** Quan hệ phân cấp giữa các khái niệm
- **Relations (R):** Các quan hệ giữa các khái niệm
- **Operators (Ops):** Tập các toán tử
- **Rules:** Các luật suy diễn
- **Objects:** Các đối tượng (instance) cụ thể

Hệ quản trị KBMS cung cấp các chức năng chính như:

- Định nghĩa tri thức
- Lưu trữ và tổ chức tri thức
- Truy vấn tri thức

Hệ thống được thiết kế theo kiến trúc nhiều tầng, kế thừa tư tưởng của các hệ quản trị cơ sở dữ liệu phổ biến nhằm đảm bảo tính mở rộng, linh hoạt và hiệu quả trong xử lý.

3.2. Các thành phần chính của KBMS

Kiến trúc của hệ quản trị cơ sở tri thức KBMS được thiết kế theo mô hình phân tầng, bao gồm bốn thành phần chính: Application Layer, KBMS Server Layer, Knowledge Storage Engine và Physical Storage.

Trong đó, Application Layer đóng vai trò cung cấp giao diện tương tác giữa người dùng và hệ thống KBMS, cho phép thực hiện các thao tác như định nghĩa và truy vấn tri thức. Tầng này bao gồm các thành phần chính như giao diện dòng lệnh (Command Line Interface - CLI), các ứng dụng khách (Client Applications) và bộ quản lý kết nối (Connection Manager). Thông qua lớp này, người dùng có thể tương tác với hệ thống bằng các câu lệnh truy vấn được thiết kế tương tự ngôn ngữ SQL, cho phép thực hiện các thao tác như xác thực người dùng, tạo và lựa chọn cơ sở tri thức, cũng như thực hiện các thao tác CRUD và truy vấn tri thức. Ví dụ, người dùng có thể đăng nhập vào hệ thống, tạo cơ sở tri thức mới, chèn dữ liệu đối tượng và thực hiện truy vấn tri thức thông qua các câu lệnh như: LOGIN admin password; CREATE KNOWLEDGE BASE geometry; USE geometry; INSERT INTO TAMGIAC VALUES (a=3, b=4, c=5); FIND TAMGIAC WHERE a=3 hoặc SOLVE TAMGIAC KNOWN ha=3 hb=4 FIND S.

Tiếp theo, KBMS Server Layer là thành phần trung tâm, chịu trách nhiệm xử lý truy vấn, phân tích cú pháp, tối ưu hóa và điều phối quá trình thực thi, đồng thời tích hợp bộ suy luận để hỗ trợ các truy vấn có yêu cầu suy diễn. Tầng này bao gồm các module chính như Connection Manager, Authentication Manager, Query Parser, Query Optimizer và Knowledge Manager. Mỗi module đảm nhận một chức năng riêng biệt, phối hợp với nhau để đảm bảo hệ thống hoạt động hiệu quả và ổn định.

(1) Module Connection Manager chịu trách nhiệm quản lý các kết nối từ client đến hệ thống KBMS. Thành phần này thực hiện việc tạo và duy trì các phiên làm việc (session), quản lý kết nối của người dùng và đóng kết nối khi cần thiết. Trong trường hợp hệ thống hoạt động ở chế độ máy chủ, các client sẽ kết nối đến KBMS thông qua giao thức TCP socket, đảm bảo khả năng giao tiếp linh hoạt giữa các thành phần.

(2) Module Authentication Manager đảm nhận chức năng xác thực người dùng trước khi cho phép truy cập vào hệ thống. Người dùng thực hiện đăng nhập thông qua các câu lệnh như LOGIN admin password, sau đó hệ thống kiểm tra thông tin xác thực và cấp quyền truy cập tương ứng. Hệ thống có thể hỗ trợ nhiều mức phân quyền khác nhau như ADMIN, DEVELOPER, USER và READONLY, nhằm kiểm soát quyền truy cập và đảm bảo an toàn dữ liệu.

(3) Module Query Parser có nhiệm vụ phân tích cú pháp các câu lệnh truy vấn do người dùng nhập vào. Quá trình này bao gồm phân tích từ vựng và cú pháp để chuyển đổi câu lệnh truy vấn thành cấu trúc cây cú pháp trừu tượng (Abstract Syntax Tree - AST), làm cơ sở cho các bước xử lý tiếp theo. Ví dụ, câu lệnh INSERT INTO TAMGIAC VALUES (a=3, b=4, c=5) sẽ được chuyển đổi thành một AST tương ứng.

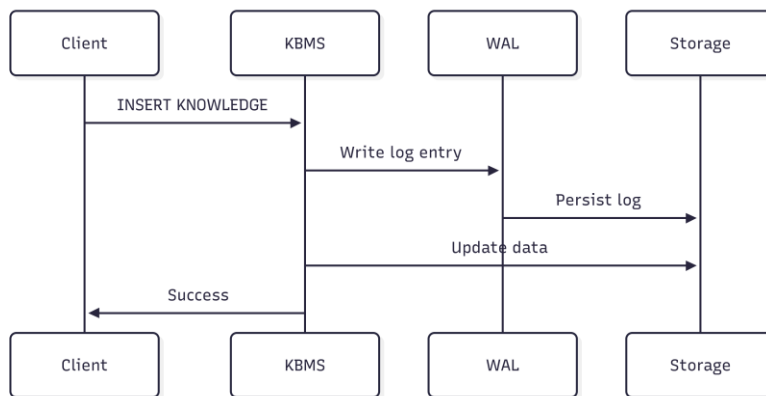
(4) Module Query Optimizer thực hiện tối ưu hóa truy vấn trước khi đưa vào thực thi. Thành phần này có nhiệm vụ lựa chọn các chiến lược xử lý hiệu quả, chẳng hạn như sử dụng chỉ mục phù hợp, giảm số lần quét dữ liệu hoặc tối ưu hóa quá trình suy luận dựa trên luật. Nhờ đó, hệ thống có thể cải thiện đáng kể hiệu năng xử lý truy vấn.

(5) Module Knowledge Manager là thành phần quản lý toàn bộ tri thức trong hệ thống KBMS, bao gồm nhiều module con như KB Manager, Object Kind Manager, Relation Manager, Operator Manager, Function Manager và Rule Manager. KB Manager quản lý các cơ sở tri thức thông qua các thao tác như tạo, lựa chọn và xóa cơ sở tri thức. Object Kind Manager quản lý các loại đối tượng tri thức, ví dụ như điểm, đoạn thẳng, góc và tam giác, trong đó mỗi loại đối tượng được định nghĩa bởi các biến, ràng buộc, quan hệ tính toán và luật liên quan. Relation Manager quản lý các quan hệ giữa các khái niệm, ví dụ như quan hệ song song, quan hệ vuông góc hoặc quan hệ phụ thuộc. Operator Manager và Function Manager lần lượt quản lý các toán tử và hàm được sử dụng trong biểu diễn tri thức, chẳng hạn như các phép toán số học hoặc các hàm lượng giác. Cuối cùng, Rule Manager quản lý các luật suy diễn, ví dụ như định lý Pitago $a^2 + b^2 = c^2$ để suy ra tam giác vuông.

Thành phần kế tiếp là **Knowledge Storage Engine** đảm nhiệm việc quản lý, tổ chức và truy xuất dữ liệu tri thức, cung cấp các cơ chế lưu trữ hiệu quả nhằm tối ưu hiệu năng hệ thống. Thành phần này bao gồm các module chính như KB Storage, Object Storage, Rule Storage, Index Manager và WAL Manager. Mỗi module đảm nhận việc lưu trữ các loại dữ liệu tri thức khác nhau, đồng thời phối hợp để đảm bảo hiệu năng truy xuất và tính toàn vẹn của dữ liệu.

- Tri thức trong hệ thống được lưu trữ dưới dạng tệp nhị phân với định dạng .kb, nhằm tối ưu hóa không gian lưu trữ và tốc độ truy xuất. Các tệp này được tổ chức theo từng loại dữ liệu, chẳng hạn như geometry.kb dùng để lưu thông tin về cơ sở tri thức, geometry_objects.kb lưu các đối tượng cụ thể và geometry_rules.kb lưu các luật suy diễn. Cấu trúc tệp được thiết kế theo dạng nhị phân có tổ chức, cho phép truy xuất nhanh và hỗ trợ mở rộng trong tương lai.

- Module Index Manager đóng vai trò tăng tốc quá trình truy vấn bằng cách xây dựng và quản lý các cấu trúc chỉ mục. Hệ thống có thể sử dụng các kỹ thuật chỉ mục phổ biến như B+ Tree hoặc Hash Index tùy theo loại dữ liệu và truy vấn. Các chỉ mục này được lưu trữ riêng trong các tệp, ví dụ INDEX.idx, nhằm hỗ trợ truy xuất dữ liệu hiệu quả mà không cần quét toàn bộ dữ liệu.

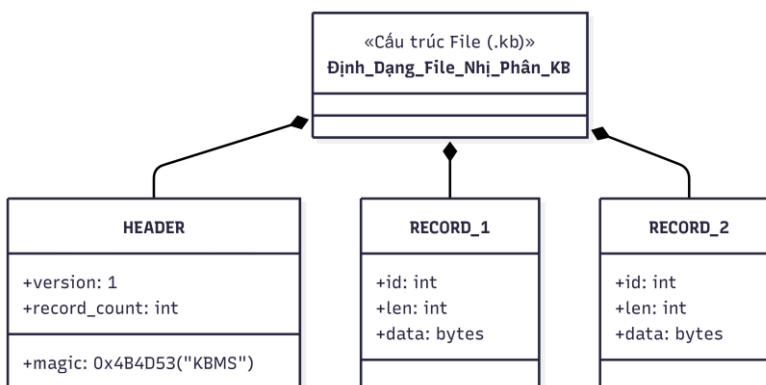


Hình 1. Ví dụ minh họa cơ chế ghi Write-Ahead Logging trong KBMS

- Module WAL Manager giúp quản lý log nhằm đảm bảo tính nhất quán và khả năng phục hồi của hệ thống. Cơ chế WAL (Write-Ahead Logging) thực hiện việc ghi log các thay đổi trước khi ghi dữ liệu thực xuống đĩa. Cụ thể, khi thực hiện một thao tác như chèn dữ liệu, hệ thống sẽ ghi thông tin vào tệp log (ví dụ WAL.log), sau đó thực hiện commit và cuối cùng mới ghi dữ liệu xuống bộ nhớ lưu trữ. Cơ chế này giúp hệ thống có thể phục hồi dữ liệu trong trường hợp xảy ra sự cố, đồng thời đảm bảo tính toàn vẹn của cơ sở tri thức.

Cuối cùng, lớp Physical Storage chịu trách nhiệm lưu trữ dữ liệu tri thức ở mức vật lý trên hệ thống tệp, đảm bảo tính bền vững và toàn vẹn của dữ liệu. Toàn bộ dữ liệu của KBMS được tổ chức trong một thư mục gốc, thường được đặt tên là data/, trong đó các thành phần tri thức được lưu trữ thành các tệp riêng biệt nhằm đảm bảo tính độc lập, dễ quản lý và thuận tiện cho việc mở rộng.

Cụ thể, mỗi cơ sở tri thức được biểu diễn thông qua nhiều tệp nhị phân tương ứng với từng loại thành phần, chẳng hạn như geometry.kb lưu thông tin tổng thể của cơ sở tri thức, geometry_kinds.kb lưu các loại đối tượng, geometry_relations.kb lưu các quan hệ, geometry_operators.kb và geometry_functions.kb lần lượt lưu các toán tử và hàm, trong khi geometry_rules.kb và geometry_objects.kb lưu các luật suy diễn và các đối tượng tính toán cụ thể. Ngoài ra, hệ thống còn sử dụng các tệp hỗ trợ như INDEX.idx để lưu trữ cấu trúc chỉ mục và WAL.log để ghi lại nhật ký thay đổi dữ liệu. Cách tổ chức lưu trữ này giúp hệ thống dễ dàng quản lý từng loại dữ liệu tri thức một cách độc lập, đồng thời hỗ trợ hiệu quả cho các cơ chế truy xuất, tối ưu hóa và phục hồi dữ liệu trong quá trình vận hành.



Hình 2. Ví dụ minh họa về cấu trúc tệp nhị phân .kb

3.3. Cấu trúc và ngôn ngữ truy vấn tri thức

3.3.1. Cấu trúc dữ liệu tri thức trong COKB

Cấu trúc dữ liệu tri thức trong mô hình COKB là thành phần nền tảng, đóng vai trò cốt lõi làm cơ sở cho việc thiết kế và xây dựng hệ quản trị cơ sở tri thức KBMS. Trước hết, cấu trúc **KnowledgeBase**

là thành phần cơ bản nhất, dùng để định danh một cơ sở tri thức (KB). Cấu trúc này bao gồm các thuộc tính như *id* (mã định danh duy nhất của KB), *name* (tên của KB, ví dụ: “geometry”, “algebra”), *created_at* (thời điểm tạo KB), cùng với các thông tin thống kê như *object_count* (tổng số lượng các thực thể - objects) và *rule_count* (tổng số lượng luật - rules) trong cơ sở tri thức.

Tiếp theo, cấu trúc ObjectKind (Loại đối tượng) đóng vai trò như một “khuôn mẫu” (blueprint) để định nghĩa thuộc tính và hành vi cho một lớp đối tượng, chẳng hạn như loại TAMGIAC. Mỗi ObjectKind bao gồm các thuộc tính như *id* và *kb_id* để định danh loại đối tượng và liên kết với cơ sở tri thức tương ứng, *name* là tên loại đối tượng (ví dụ: “TAMGIAC”, “DUONG_TRON”), *variable_count* và *variables* thể hiện số lượng và danh sách các biến thành phần (ví dụ: các cạnh a, b, c), *constraint_count* và *constraints* mô tả các ràng buộc toán học nội tại (ví dụ: tổng hai cạnh lớn hơn cạnh còn lại), *comp_rel_count* và *comp_rels* biểu diễn các quan hệ tính toán (computation relations) cho phép suy ra giá trị giữa các biến, cùng với *fact_count/facts* và *rule_count/rules* tương ứng với các sự kiện mặc định và các luật áp dụng riêng cho loại đối tượng này.

Đối với các toán tử và hàm, hệ thống sử dụng hai cấu trúc OperatorDef và FunctionDef để định nghĩa. Hai cấu trúc này có đặc điểm tương đồng, được sử dụng để mô tả các toán tử và hàm phục vụ cho quá trình tính toán tri thức. Cụ thể, *symbol* (chỉ áp dụng cho Operator) biểu diễn ký hiệu toán tử như +, -, *, /, ^, =; *param_count* và *params* xác định số lượng và kiểu dữ liệu của các tham số đầu vào; *result_type* là kiểu dữ liệu trả về (ví dụ: REAL, BOOLEAN); *properties* mô tả các đặc tính bổ sung như tính giao hoán hoặc kết hợp; và *procedure* là chuỗi mô tả cách thức thực hiện phép toán hoặc hàm tương ứng.

Cấu trúc Rule (Luật suy diễn) được sử dụng để định nghĩa các quy tắc logic nhằm cho phép hệ thống suy luận ra sự kiện mới từ các sự kiện đã biết. Mỗi luật bao gồm *kind_rule* (loại hoặc tên phân loại của luật), *hypothesis_count* và *hypothesis* biểu diễn phần giả thiết (If...), và *conclusion_count* cùng *conclusion* biểu diễn phần kết luận (Then...).

Ở mức dữ liệu cụ thể, các thực thể được biểu diễn thông qua cấu trúc ObjectInstance, là các thể hiện (instance) của một ObjectKind. Cấu trúc này bao gồm *object_type* (tên loại đối tượng, ví dụ: “TAMGIAC”), *params* là các thông tin định danh hoặc tham số khởi tạo, và *variable_count* cùng *values* biểu diễn số lượng biến và giá trị cụ thể của chúng (ví dụ: a=3, b=4, c=5), trong đó các giá trị này được lưu trữ dưới dạng nhị phân trong mảng *values* nhằm tối ưu hiệu năng.

Cuối cùng, cấu trúc QueryResult (Kết quả truy vấn) được sử dụng để trả về kết quả cho người dùng sau khi thực hiện các lệnh như FIND hoặc SOLVE. Cấu trúc này bao gồm *success* (trạng thái thực thi thành công hay không), *message* (thông báo phản hồi, bao gồm lỗi hoặc xác nhận), *objects* (danh sách các thực thể thỏa điều kiện truy vấn) và *derived_values* là một bản đồ (map) chứa các giá trị mới được suy luận ra, chẳng hạn như diện tích S được tính từ các cạnh của tam giác.

3.3.2. Ngôn ngữ truy vấn tri thức trong KBMS

Ngôn ngữ truy vấn của hệ quản trị cơ sở tri thức (KBMS) được thiết kế dựa trên tư duy của SQL trong các hệ quản trị cơ sở dữ liệu truyền thống, đồng thời được mở rộng nhằm hỗ trợ biểu diễn và suy luận tri thức. Ngôn ngữ KBMS cho phép người dùng định nghĩa, thao tác và truy vấn các đối tượng tri thức có ngữ nghĩa, cũng như thực hiện suy luận dựa trên các luật đã được xây dựng. Về tổng thể, ngôn ngữ truy vấn KBMS được tổ chức thành ba nhóm chính: KDL (Knowledge Definition Language) dùng để định nghĩa cơ sở tri thức và các thành phần tri thức; KML (Knowledge Manipulation Language) dùng để thao tác trên các thực thể tri thức; và KQL (Knowledge Query Language) dùng để truy vấn và khai thác tri thức.

Nhóm ngôn ngữ định nghĩa tri thức KDL cho phép người dùng tạo và quản lý cơ sở tri thức, bao gồm các lệnh như tạo cơ sở tri thức, lựa chọn môi trường làm việc hoặc hiển thị các thành phần tri thức. Nhóm ngôn ngữ thao tác KML hỗ trợ thao tác trên các đối tượng tri thức dưới dạng thực thể có cấu trúc, cho phép thêm hoặc cập nhật các đối tượng với các thuộc tính cụ thể, thay vì các bản ghi dữ liệu

thuần túy. Nhóm ngôn ngữ truy vấn và khai thác tri thức KQL cung cấp khả năng truy vấn các đối tượng tri thức dựa trên điều kiện, kết quả trả về là các đối tượng tri thức có ngữ nghĩa được biểu diễn theo mô hình ontology COKB.

Bảng 1. Danh mục nhóm ngôn ngữ truy trong KBMS

Danh mục KBMS	Các dạng yêu cầu xử lý trong ngôn ngữ truy vấn	Ví dụ minh họa
Knowledge Definition Language (KDL)	Định nghĩa cấu trúc tri thức như tạo, quản lý CSTT, khai báo Concepts, Hierarchy, Relations, Operators, Rules, Objects...	CREATE KNOWLEDGE BASE, CREATE CONCEPT, CREATE RULE, CREATE HIERARCHY, CREATE RELATION
Knowledge Manipulation Language (KML)	Thêm, sửa, xóa và xử lý các sự kiện hoặc các objects với các thuộc tính cụ thể.	INSERT, UPDATE, DELETE, IMPORT, EXPORT
Knowledge Query Language (KQL)	Truy vấn và khai thác tri thức.	SELECT, EXPLAIN, DESCRIBE
Nhóm lệnh thao tác khác	Các thao tác quản trị trên KBMS.	LOGIN, EXIT, GRANT, REVOKE, SHOW

3.3.3. Thuật toán xử lý truy vấn trong KBMS

Algorithm 3.3.1: Query Processing in KBMS

Input: query

Output: result

```

1: tokens ← Phân tích xử lý từ vựng từ query
2: if tokens is invalid then
3:   return "Lexical Error"
4: end if
5: ast ← Xây dựng AST tree từ tokens
6: if ast is invalid then
7:   return "Syntax Error"
8: end if
9: plan ← Tối ưu hóa cây AST
10: data ← Truy xuất dữ liệu từ hệ lưu trữ
11: result ← Áp dụng bộ lọc trên data
12: trả về result

```

Thuật toán có thể mô tả thành 5 bước cơ bản như sau:

- **Bước 1 - Lexical Analysis:** Ở bước này, truy vấn đầu vào được xử lý bởi bộ phân tích từ vựng nhằm chuyển đổi chuỗi ký tự thành danh sách các token. Hệ thống quét tuần tự từng ký tự hoặc nhóm ký tự để nhận diện các thành phần như từ khóa, định danh, toán tử và giá trị. Kết quả của bước này là một chuỗi token đã được chuẩn hóa, làm đầu vào cho giai đoạn phân tích cú pháp.

- **Bước 2 - Parsing:** Danh sách token được đưa vào bộ phân tích cú pháp để kiểm tra tính hợp lệ của truy vấn theo đặc tả EBNF. Nếu truy vấn hợp lệ, hệ thống sẽ xây dựng cây cú pháp trừu tượng (AST), biểu diễn cấu trúc logic của truy vấn. AST đóng vai trò trung gian giữa cú pháp và thực thi, tương tự như query tree trong các hệ quản trị cơ sở dữ liệu.

- **Bước 3 - Query Optimization:** Từ cây AST, hệ thống sinh ra kế hoạch thực thi (execution plan). Ở bước này, các kỹ thuật tối ưu được áp dụng nhằm giảm chi phí xử lý, chẳng hạn như đẩy điều kiện lọc xuống gần nguồn dữ liệu, loại bỏ các phép toán dư thừa và lựa chọn chiến lược thực thi phù hợp.

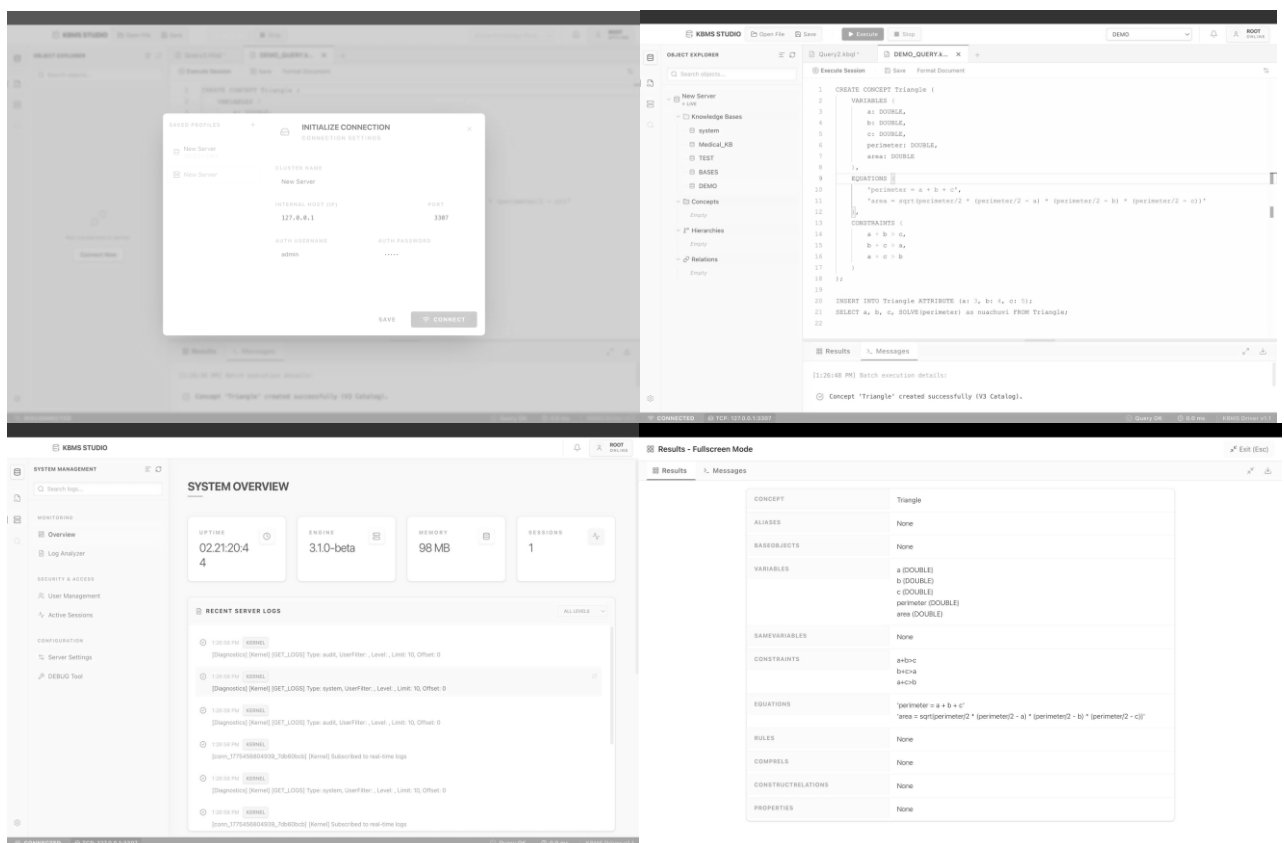
Kết quả là một kế hoạch thực thi hiệu quả hơn so với biểu diễn ban đầu.

- Bước 4 - Execution: Kế hoạch thực thi được chuyển xuống tầng thực thi để truy xuất dữ liệu từ hệ lưu trữ. Hệ thống thực hiện các thao tác như đọc dữ liệu, lọc theo điều kiện và tính toán nếu cần. Đây là giai đoạn tiêu tốn nhiều tài nguyên nhất và tương ứng với execution engine trong DBMS truyền thống.

- Bước 5 - Result Generation: Sau khi thực thi xong, kết quả được xử lý và trả về cho người dùng. Kết quả có thể được định dạng lại tùy theo yêu cầu. Trong KBMS, bước này cũng có thể kết hợp với cơ chế suy luận để bổ sung tri thức nếu truy vấn yêu cầu.

4. KẾT QUẢ VÀ BÀN LUẬN

Trong quá trình thực hiện xây dựng mô hình đề xuất, một hệ quản trị cơ sở tri thức (KBMS) dạng đơn giản đã được xây dựng và cung cấp giao diện sử dụng dưới dạng ứng dụng web. Hệ thống này cho phép người sử dụng tương tác trực tiếp với cơ sở tri thức thông qua giao diện thân thiện, hỗ trợ các chức năng cơ bản như tạo lập, khai báo và truy vấn tri thức. Cụ thể, người dùng có thể xây dựng cơ sở tri thức theo mô hình COKB bằng cách định nghĩa các thành phần như khái niệm, đối tượng, thuộc tính, quan hệ, sự kiện và luật. Bên cạnh đó, hệ thống cung cấp ngôn ngữ truy vấn được thiết kế trong bài báo, cho phép người dùng thực hiện các thao tác truy vấn dữ liệu cũng như truy vấn tri thức một cách linh hoạt. Môi trường thực nghiệm triển khai KBMS trên server với nền tảng .NET 8 SDK và NodeJS v18+.



Hình 3. Giao diện hỗ trợ người dùng quản lý và thao tác trên KBMS

Thông qua ứng dụng web, các câu lệnh truy vấn được xử lý theo quy trình đã đề xuất, bao gồm phân tích cú pháp, tối ưu hóa và thực thi, đồng thời hỗ trợ cơ chế suy luận khi cần thiết. Kết quả truy vấn được hiển thị trực quan, giúp người dùng dễ dàng khai thác và kiểm chứng tri thức. Việc hiện thực hệ thống dưới dạng ứng dụng web không chỉ minh họa tính khả thi của mô hình KBMS mà còn cho thấy khả năng ứng dụng thực tiễn trong các hệ thống hỗ trợ học tập, hệ chuyên gia và các nền tảng

quản lý tri thức hiện đại.

Bảng 2. Danh mục kiểm thử hiệu năng KBMS

Nhóm	Danh mục kiểm thử	Phân loại	Kết quả
001	Performance Benchmarks	Stress Testing	Passed
002	Storage Architecture	Scalability Testing	Passed
003	Transactions & WAL	Scalability Testing	Passed
004	Query Engine	Scalability Testing	Passed
005	Schema Evolution	Scalability Testing	Passed

Hệ thống KBMS được triển khai dựa trên kiến trúc đề xuất đã đạt được hiệu năng với thông lượng tối đa 200,000+ ops/sec và độ trễ truy vấn < 10 ms. Tuy nhiên, hệ thống hiện tại chưa hỗ trợ đầy đủ cơ chế suy luận sau khi khai báo tri thức do chức năng này chưa được hoàn thiện trong giai đoạn kiểm thử. Bên cạnh đó, kiến trúc hệ thống mới chỉ được cài đặt ở dạng standalone, chưa triển khai theo mô hình phân tán, do đó các khả năng như tính HA (high availability) và mở rộng (scalability) khi khối lượng tri thức tăng lớn vẫn chưa được khai thác đầy đủ.

5. KẾT LUẬN

Bài báo đã đề xuất một mô hình hệ quản trị cơ sở tri thức (KBMS) dựa trên nền tảng mô hình ontology COKB, nhằm giải quyết bài toán quản lý và khai thác tri thức một cách có cấu trúc và hiệu quả. Khác với các hệ quản trị cơ sở dữ liệu truyền thống, KBMS không chỉ hỗ trợ lưu trữ và truy vấn dữ liệu mà còn tích hợp chức năng quản trị, truy vấn và suy luận dựa trên tri thức.

Bài báo trình bày chi tiết kiến trúc hệ thống, xây dựng một kiến trúc đa tầng cho KBMS, bao gồm các thành phần từ tầng giao tiếp, xử lý truy vấn, đến lưu trữ và suy luận. Ngôn ngữ truy vấn được thiết kế theo hướng kế thừa SQL nhưng cải tiến thêm thành phần truy vấn tri thức và suy luận, giúp hệ thống vừa dễ tiếp cận vừa có khả năng biểu diễn tri thức phức tạp. Kết quả cho thấy mô hình KBMS có khả năng mở rộng tốt, phù hợp với các bài toán yêu cầu xử lý tri thức được biểu diễn theo mô hình COKB, đặc biệt trong các lĩnh vực như giáo dục, hệ chuyên gia và hệ hỗ trợ ra quyết định.

Trong tương lai, nghiên cứu có thể được mở rộng theo hướng tối ưu hóa hiệu năng suy luận thông qua các kỹ thuật như indexing cho luật, caching kết quả suy luận và heuristic search để tối ưu hóa truy xuất vào lớp Storage Layer. Đồng thời phát triển ngôn ngữ truy vấn với các cấu trúc nâng cao như truy vấn lồng (nested query), truy vấn theo ngữ nghĩa (semantic query) và tích hợp các biểu thức logic phức tạp.

LỜI CẢM ƠN

Nghiên cứu này được trường Đại học Quốc tế Hồng Bàng cấp kinh phí thực hiện dưới mã số đề tài GVTC19.03.

TÀI LIỆU THAM KHẢO

- [1] N. V. Do, "Ontology COKB for designing knowledge-based systems," in *Proceeding of 13th International Conference on Intelligent Software Methodologies, Tools, and Techniques*, 2014.
- [2] Hien D. Nguyen, Nhon V. Do, Vuong T. Pham, "A method for designing Knowledge-based Systems and Application," in *Applications Computational Intelligence Multi-Disciplinary Research*, Elsevier, 2022, pp. 159-185.
- [3] Nhon Do, Thanh T. Mai, "Perfect COKB Model and Reasoning Methods for the design of Intelligent Problem Solvers," in *International Conference on Intelligent Software Methodologies, Tools, and Techniques*, Japan, 2017.
- [4] Đ. V. Nhon, "Các hệ thống trí tuệ nhân tạo ứng dụng trong giáo dục," *Hong Bang International*

University Journal of Science, 2023.

[5] Diem Nguyen, "Intelligent Problem Solving about Functional Component of COKB Model and Application," in *International Conference on Computational Collective Intelligence Technologies and Applications*, 2014.

[6] Phat Huynh, Nhon V. Do, "A Variation of COKB Model for Solving Problems about Chemical Elements," in *International Conference on Knowledge and Systems Engineering*, 2015.

[7] Nhon V. Do, Thanh T. Mai, Long N. Hoang, "A Knowledge-Based Model for Designing the Knowledge Querying System in Education," in *International Conference on Research, Innovation and Vision for the Future*, Ho Chi Minh, 2022.

[8] Nhon V. Do, Thanh T. Mai, "A Knowledge Representation Model for Designing the Knowledge Querying System in Programming Language C/C++," in *International Conference on Computing and Communication Technologies (RIVF)*, 2023.

[9] Trúc Ly, "Xây dựng hệ chuyên gia hỗ trợ chẩn đoán và điều trị các biến chứng trên mạch máu nhỏ của bệnh đái tháo đường," Luận văn thạc sĩ, Hồ Chí Minh City, 2013.

[10] Franz Baader, Technische Universität, Dresden, Ian Horrocks, University of Oxford, Carsten Lutz, Universität Bremen, Uli Sattler, University of Manchester, *An Introduction to Description Logic*, Cambridge University Press, 2017.

[11] C. M. Keet, *An Introduction to Ontology*, Creative Commons Attribution 4.0 International License (CC BY 4.0), 2018.

[12] Research, Stanford Center for Biomedical Informatics, "Protégé," The Stanford University School of Medicine, The Stanford University School of Medicine, 2025.

[13] Inc, About Franz, "TopBraid Composer," About Franz Inc.

[14] Ontotext, "GraphDB," Ontotext, 2023.

[15] Group, ART, "VocBench," Università degli Studi di Roma Tor Vergata, Roma, 2023.