Hindawi
Applied Computational Intelligence and Soft Computing
Volume 2020, Article ID 9834218, 14 pages
https://doi.org/10.1155/2020/9834218

Research Article

Some Criteria of the Knowledge Representation Method for an Intelligent Problem Solver in STEM Education

Hien D. Nguyen,[1,2] **Nhon V. Do**,[3] **Nha P. Tran**,[4] **Xuan Hau Pham**,[5] **and Vuong T. Pham**[6]

[1]*University of Information Technology, Ho Chi Minh City, Vietnam*
[2]*Vietnam National University, Ho Chi Minh City, Vietnam*
[3]*Ho Chi Minh City Open University, Ho Chi Minh City, Vietnam*
[4]*University of Transport and Communications, Campus in Ho Chi Minh City, Hanoi, Vietnam*
[5]*Quang Binh University, Đồng Hới, Dong Hoi City, Quang Binh, Vietnam*
[6]*Sai Gon University, Ho Chi Minh City, Vietnam*

Correspondence should be addressed to Hien D. Nguyen; hiennd@uit.edu.vn and Nha P. Tran; tpnha@utc2.edu.vn

Received 1 July 2019; Revised 16 November 2019; Accepted 3 December 2019; Published 13 May 2020

Academic Editor: Aniello Minutolo

Nowadays, building intelligent systems for science, technology, engineering, and math (STEM) education is necessary to support the studying of learners. Intelligent problem solver (IPS) is a system that can be able to solve or tutor how to solve the problems automatically. Learners only declare hypothesis and goal of problems based on a sufficient specification language. They can request the program to solve it automatically or to give instructions that help them to solve it themselves. Knowledge representation plays a vital role in these kinds of intelligent systems. There are various methods for knowledge representation; however, they do not meet the requirements of an IPS in STEM education. In this paper, we propose the criteria of a knowledge model for an IPS in education. These criteria orient to develop a method for knowledge representation to meet actual requirements in practice, especially pedagogical requirements. For proving the effectiveness of these criteria, a knowledge model is also constructed. This model can satisfy these criteria and be applied to build IPS for courses, such as mathematics and physics.

1. Introduction

Knowledge representation plays a vital role in designing intelligent systems. Science, technology, engineering, and math (STEM) education emphasizes connections about concepts across different STEM fields to treat STEM education as a whole [1]. STEM education equips knowledge science and their real-world applications for the students. Then, the students can develop their ability for discovering and problem-solving.

The circle of STEM education is described in Figure 1. "Science" in the STEM circle means the process of scientific innovation from "technology" to "knowledge." In practice, when meeting the technology, scientists always make questions for researching to complete the technology. When finding the solutions for those questions, they will invent new scientific knowledge.

In contrast, "engineering" in the STEM circle uses the scientific knowledge to design new technologies [1, 2]. The engineers have to solve problems to apply scientific knowledge to the practice. Science is the scientific process to invent new knowledge, and engineering is also the technical process to create new technologies. Two processes combine to form the scientific and technical innovation cycle, which has a spiral shape. After every turn of this spiral procedure, scientific knowledge improves with the development of the new technologies [2].

Courses of STEM education mentioned in this paper are mathematics, natural sciences (such as physics and chemistry), and basic programming (such as introduction to programming, data structures, and algorithms).

Artificial intelligence applications can be used to support the practice of learning. Three promising applications are intelligent tutoring systems, automated essay scoring, and

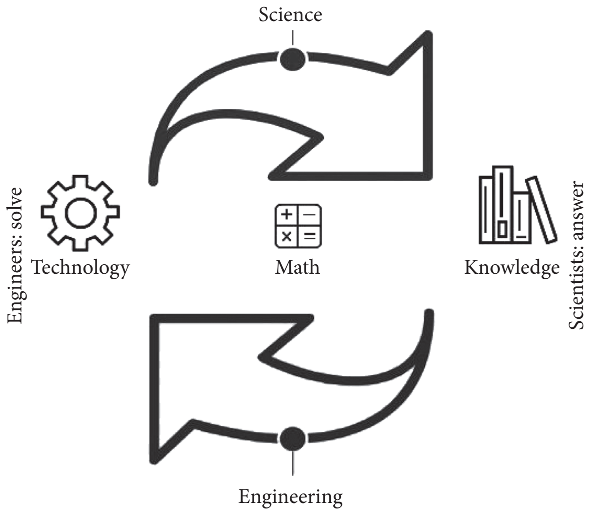

Figure 1: The STEM circle.

early warning systems [3]. An intelligent tutoring system (ITS) simulates the instructional experience and interactions between a learner and a human tutor [4]. An intelligent problem solver (IPS) is a part of the ITS that can solve the problems automatically. Learners only declare the hypothesis and goal of problems based on a sufficient specification language [5]. They can request the program to solve it automatically or to give instructions that help them to solve it themselves. The architecture of an IPS in education is shown in Figure 2.

The primary process of IPS is as follows: Through the user interface, the system recognizes the problems which are specified by the suitable specification language. The hypothesis and goal of the problem are determined by analysing the inference engine, and then they will be recorded into the working memory. The system uses its knowledge base to search objects, facts, and rules. After that, the system uses reasoning rules to solve the problem. When the system finds the solution, it produces a good one. Finally, this right solution is output in a human-readable form for the users via the interface.

For supporting the studying of learners, an IPS in STEM education has to meet requirements [7, 8] as follows:

RQ1: the program can solve common exercises in the course. Based on the knowledge base, the system can solve the basic and advanced kinds of general problems in the curriculum of the course automatically.

RQ2: the input problems are specified by the language similar to the human. The solutions of the program are readable, step-by-step.

RQ3: the reasoning of this system uses the knowledge of the learner about the course. Its solutions are similar to the solving method of the student.

For meeting these requirements, the method for knowledge representation in this system has to be built based on specific criteria. The knowledge base of this system is sufficient. It has to be organized fully and exactly. This thing will meet RQ1 of an IPS system. Besides, for satisfying RQ2, the knowledge representation method in this system is also convenient for users. The users can understand the methods for solving exercises.

Moreover, the problem-solving reasoning simulates the way of the learner's thinking. The reasoning steps are suitable for the knowledge level of the learner. Therefore, this program satisfies requirements RQ2 and RQ3 of an IPS in education.

In this paper, we propose the criteria of a method to represent the knowledge of an IPS in STEM education. The method for knowledge representation includes a knowledge model, model of problems, and reasoning method to solve problems. For proving the effectiveness of these criteria, a knowledge model has been presented in this paper. This model can represent the knowledge of relations and operators, called the Rela-Ops model. This model satisfies the criteria of a knowledge model for an IPS in education.

The criteria in this study develop a representation method about theory and application. Each criterion has also been classified into certain levels. They have to guarantee the following factors:

(i)Theory: these criteria tend to ensure the solid foundation of the knowledge representation method. They make this method developing in-depth.

(ii)Application: these criteria help to build a knowledge model which can be appllied in the real world, especially knowledge domains in STEM education, such as mathematics, physics, and chemistry. This model can be used to design the inference engine for solving practice problems of knowledge domains. The solutions are readable, step-by-step, and reasoning steps are suitable for the knowledge level of learners.

The Rela-Ops model is built based on the ontology approach [9]. It is useful in designing IPS in courses of STEM, such as discrete mathematics in university, solid and plane geometry in high-school, vector algebra in high-school, and direct current electrical circuit in middle-school. This model was presented in [9]. In this paper, we build the method or the process to design the IPS in education which can satisfy requirements of the IPS. Based on the application of this process, we also introduce the Rela-Ops model in general to prove the effectiveness of the proposed criteria.

The next section discusses related studies for the criteria of a method for knowledge representation of IPS in education. Section 3 proposes and specifies the criteria of a knowledge representation method for an IPS in education. Section 4 presents a method for designing an IPS in education. The knowledge representation of this method satisfies the proposed criteria. Section 5 presents the Rela-Ops model, which is a knowledge model of relations and operator. This model meets the criteria of a knowledge model for an IPS in education. Section 6 discusses and makes a comparison between knowledge representation methods. The last section concludes the results.

2. Related Work

There are many studies for the criteria of a method for knowledge representation. However, the criteria discussed

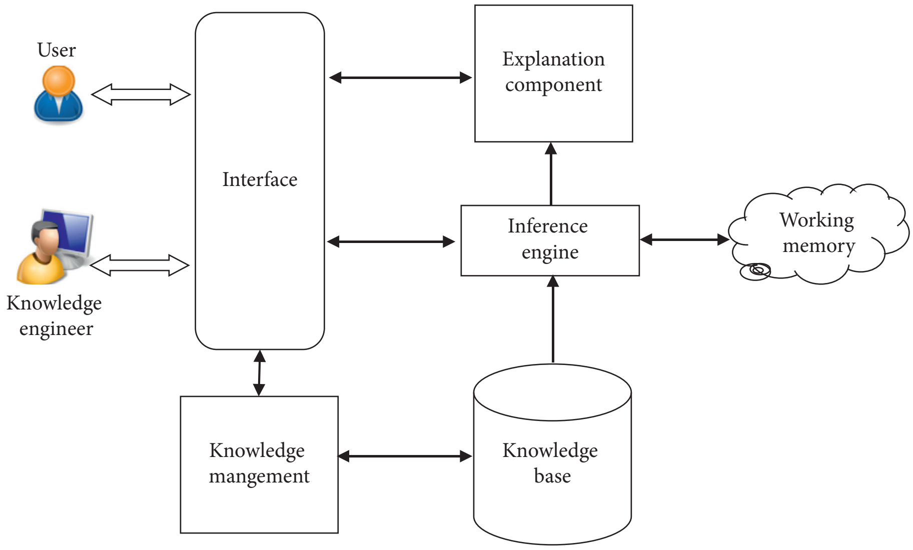

Figure 2: The architecture of an IPS in education [5, 6].

in previous works do not meet the requirements of an IPS in STEM education.

A knowledge model has to represent the practical knowledge domain adequately [10]. It can perform fundamental components of this domain: concepts, relations, and inference rules. This representation method also gives reasoning in the knowledge [11]. Nonetheless, these criteria were still general, and they did not explain how the presentation is adequate; thus, they cannot be applied in practice, especially in building the IPS in education. The result in [12] uses belief rules based on knowledge representation scheme and inference methodology using evidential reasoning rule for representing the uncertainty of knowledge and reasoning.

Besides that, a knowledge model needs to be formal [13]. The components in the model have a solid foundation. The results in [14] study a method for generating a formal ontology by deep learning. The logic-based knowledge also can be represented by linear algebra [15]. Operators and relations in this knowledge are computed based on matrices and tensors. Nonetheless, they are theoretical results and have not yet been applied in STEM knowledge domains.

Solving system of linear equations is a critical problem in linear algebra. The study in [16] presents iterative methods for solving a linear interval system of equations, which is a linear system involving uncertain coefficients appearing as interval numbers. The first method replaces real operations with interval operations based on the conjugate gradient method. The second method solves linear interval systems by using the steepest descent idea. However, those results were not suitable to support the learning of linear algebra in university. They did not use the knowledge of the course to solve the system of linear equations.

In [17], the study presents the performance of a steam turbine in thermal power plants using an artificial neural network. This method used NARMA to generate data and train network for the controlling model. Although the results of this method are emerging, it cannot show by itself how it works. Hence, it cannot be used to train the user to understand its performance.

In [18], the authors presented some components in intelligent tutoring systems. Those components have to satisfy educational criteria to tutor the students. However, those criteria still belong to the scenarios, and they cannot be used to develop the system in practice, especially for IPS systems. In [19, 20], the authors also proposed the requirements of knowledge representation for ITS. The domain knowledge module can represent structural and relational knowledge. This representation is natural, ease of update. The inference engine is efficiency, and it can reach conclusions from partially known inputs. These requirements are practical for an ITS. An IPS is a part of an ITS which can solve problems automatically; however, these requirements do not mention to the ability about problem-solving, and thus they have some points which are not appropriate to an IPS.

A set of criteria for software requirements specification had been proposed in [21]. Those criteria are used to evaluate such standards, according to the unique characteristics of organizations and software development projects. However, those criteria are used for industrial software development, and they are not suitable for intelligent systems in education. They do not mention characteristics in studying, such as naturalness and pedagogy. Criteria of an IPS in education can combine criteria for software requirements specification in [21] to become standards for designing of knowledge representation method for IPS in general.

The paper [7] presented criteria for knowledge representation method of an IPS in education. Those criteria are: universality, usability, practicality, and formality. In this paper, some criteria are revised: generality, usability, naturalness, and formality. Each criteria is explained more clearly in each level. Those revisions make the evaluation of a knowledge representation method more easy and more suitable with the practical applications.

Table 1 summarizes current criteria of knowledge representation for IPS in education and their novelty in this paper.

3. Criteria of a Knowledge Representation Method for an Intelligent Problem Solver in Education

For meeting requirement RQ1 of an IPS in STEM education, a knowledge model has to represent the knowledge base sufficiently. This method also has a useful specification language for users and supplies a pedagogical solution of an exercise as the knowledge level of learners [13]. It makes the IPS system satisfying requirement RQ2. Besides that, the reasoning of this method simulates the reasoning of humans, especially the learner. It works based on the specified knowledge as the content of the course. The reasoning meets requirement (RQ3). Moreover, the method for knowledge representation can apply in many knowledge domains, especially in knowledge of courses. It also ensures a solid mathematical foundation [13]. Besides, the IPS in education is also an intelligent software, so it has to satisfy some selection criteria for software requirements specification standard: generality, completeness, precision, practicality, and integration [21]. For these reasons, the criteria of a knowledge representation method for an IPS in STEM education include generality, usability, naturalness, and formality.

3.1. Generality. The generality criterion examines how suitable a knowledge representation method is for different knowledge domains of courses [9]. Most of the current methods are only designed for specific types of knowledge domains [7, 19]. This criterion means a method can apply in many knowledge domains of courses.

The generality criterion provides the flexibility of the representation method. This method can be applied to represent the knowledge of courses, especially the courses about science and technology: mathematics, physics, and chemistry. When representing the practical knowledge, this method can be used directly or only needs some minor improvements for representing. This criterion includes four levels corresponding from very bad to very good. The meaning of each level is shown in Table 2:

3.2. Usability. This criterion is the completeness criterion for intelligent software requirements specification standards. The first aspect of the usability criterion is the completeness of the knowledge model. The knowledge base of the IPS includes the knowledge, the content, and the actual learning content in the curriculum. The second aspect of the usability criterion is the completeness of the reasoning. The reasoning of this knowledge model uses detailed knowledge to solve practical problems, especially joint exercises in courses completely. Moreover, the reasoning steps of solutions are as the solving method of a student.

To achieve these goals, the knowledge model has an adequate structure to represent the practical knowledge domain [20]. The human knowledge domain has numerous components, but it has a foundation, including concepts of the knowledge domain, relations between concepts, and inference rules [5, 6]. A model represents these components making a knowledge kernel as ontology. From that, the kernel can integrate with other knowledge, such as operators and functions, to strengthen the ability for representing the practical knowledge. Hence, the representation method needs the ability to represent the knowledge kernel. The inference strategy uses heuristics rules in its processing. Some heuristics rules can be used: arranging the order of rules in priority and using sample problems [25]. It can solve many kinds of exercises in the course.

Levels of the usability criterion are shown in Table 3:

3.3. Naturalness. The naturalness criterion is the practicality criterion of software. An IPS tends to two main users: knowledge engineers and learners [5, 7]. The representation method has to guarantee the specification language and the method of reasoning to accomplish the naturalness criterion.

The system has a knowledge base, which can be updated by the knowledge engineer. The specification language of the knowledge model has a simple structure but can represent the knowledge domain adequately. The representation is naturalness. Users as knowledge engineers can employ it to represent or update the knowledge domain easily.

Besides, the direct users of this system are learners. The method to input a problem into the system is easy to use by learners. The reasoning method of this model also simulates the reasoning for solving problems of the learners. They can understand the knowledge as solutions for practical exercises. The system can find the pedagogical solutions of the exercises; the reasoning for problem-solving supports learners for studying the corresponding course.

Levels of the naturalness criterion are shown in Table 4:

3.4. Formality. The formality criterion ensures the correctness of the representation method. This criterion supports theoretical shreds of evidence for the effectiveness of the method [10]. Moreover, by the formality of the method, it can be improved and developed based on the solid foundation.

Firstly, the components of the knowledge model need to be constructed based on a solid theoretical foundation [13]. Their structure and relationships are built formally. The problems of this model can also be modeled. Secondly, the algorithms for solving the problems must be constructed based on the structure of the knowledge model and problems. Those algorithms must be proven to be finite and active, and their complexity must be evaluated.

Levels of the formality criterion are shown in Table 5:

4. Method for Designing the Intelligent Problem Solver in Education

For satisfying requirements of the IPS in education, the building knowledge base and inference engine components are essential in designing the system. The knowledge

Table 1: Current criteria of knowledge representation for IPS.

Criteria	Current	Novelty in this paper
Usability (completeness)	This criterion is the completeness criterion for intelligent software requirements specification standards. It concerns the requirement for building knowledge bases of intelligent systems. The knowledge representation method can represent the components of ITS entirely by using this criterion. However, the current meaning of this criterion does not aim to design the IPS.	In practice, a knowledge domain has many levels, especially the educational knowledge. In this study, the criteria have been classified into levels. Each level has the meaning of being suitable with requirements for designing the knowledge base of each IPS
Formality	The meaning of the criterion only orients to build formal models, and it did not mention the application in real-world knowledge domain (i) *Formal logic methods* are proper to meet this criterion [22, 23]. However, those methods cannot represent real-world knowledge, especially the knowledge of courses (ii) *Algebraic approach* is a method based on the mathematical structures. They are classical algebraic structures [24]. However, this criterion does not mention the ability of reasoning and explaining in the problem-solving process	Research the criterion being suitable to apply in practice and ensure the theoretical foundation. This criterion includes: (i) Criteria about theoretical foundation for constructing components of the knowledge model. The structure of those components can be used to design algorithms for reasoning (ii) Criteria can be used to build practical, intelligent systems, especially for IPS in courses
Set of criteria for software requirements specification	Those criteria are used to evaluate standards according to the unique characteristics of specific combinations of software development projects [21] However, those criteria are not suitable for the characteristics of intelligent educational software: naturalness and pedagogy	Build the criteria for software development to adapt to the pedagogical criteria of the intelligent learning system.

Table 2: Levels of the generality criterion.

Level 1	Level 2	Level 3	Level 4
This method is only applied in a testing knowledge domain	This method is built for a particular knowledge domain	This method can be applied to knowledge domains that have specific characteristics, including concepts and relations between concepts and inference rules	This method can be applied to practical knowledge domains in education, especially in IPS systems

Table 3: Levels of usability criterion.

Level 1	Level 2	Level 3	Level 4
(i) This method is not adequate to apply in a practical knowledge domain (ii) The reasoning for this method is machinery; it is not natural	(i) This method is built for an educational knowledge domain (ii) The system can only solve some classes of exercises as frames in a course	(i) This method can represent essential components of a knowledge model in an IPS: concepts, relations, and inference rules (ii) The reasoning of the representation method can solve common exercises in the course	(i) This method can represent a knowledge model in an IPS system completely (ii) The reasoning of the system uses heuristics rules to solve problems (iii) The system can solve common exercises in a course. It can also solve some hard problems which need to combine the knowledge of the course for solving them

representation method has to meet the criteria in Section 3. The process of analysis and design of the system components consists of seven stages (Figure 3) [5].

Stage 1 collects the real knowledge domain based on the classification of kinds of knowledge. The collection helps to form the model for knowledge representation. Stage 2 builds the knowledge model for the collected knowledge domain. Based on the knowledge model, stage 3 organizes the knowledge base for the IPS. The specification language for the knowledge base, which is studied in stage 4, has to simulate the way of describing the knowledge in practice. Besides the knowledge base, the model of problems on the knowledge domain also has to be studied. Those problems are the foundation for designing of reasoning algorithms. The reasoning algorithms are the demonstration of the problem-solving ability of the system. Stage 5 designs the query language of the system. For the goal supporting of the learning, the query language has to be suitable for the

Table 4: Levels of naturalness criterion.

Level 1	Level 2	Level 3	Level 4
(i) The specification language is machinery (ii) The representation method cannot solve the common exercises of a course	(i) The specification language of the method simulates the human language, but it is not suitable for students (ii) The representation method can solve common exercises of a course	(i) The specification language of the method is suitable for students (ii) The system can solve some general problems. The reasoning of its solution is suitable for the level of learners	(i) The specification language of the method is similar to the natural language for knowledge representation (ii) Solutions of the system are pedagogy (iii) Besides solving problems automatically, the knowledge base of this system tends to tutor the student on how to solve a problem in the course

Table 5: Levels of the formality criterion.

Level 1	Level 2	Level 3	Level 4
(i) This method has not yet built based on a solid mathematical foundation (ii) It has not yet had a model of general problems in the knowledge domain	(i) This method is built based on a particular mathematical structure (ii) This method has a model of general problems based on its knowledge model	(i) This method is built based on the solid mathematical structure (ii) Problems can be modeled, and algorithms for solving them are designed based on the knowledge model	(i) This method is built based on the solid mathematical structure (ii) Problems can be modeled based on the knowledge model (iii) The finiteness and effectiveness of the algorithms for solving those problems are proved

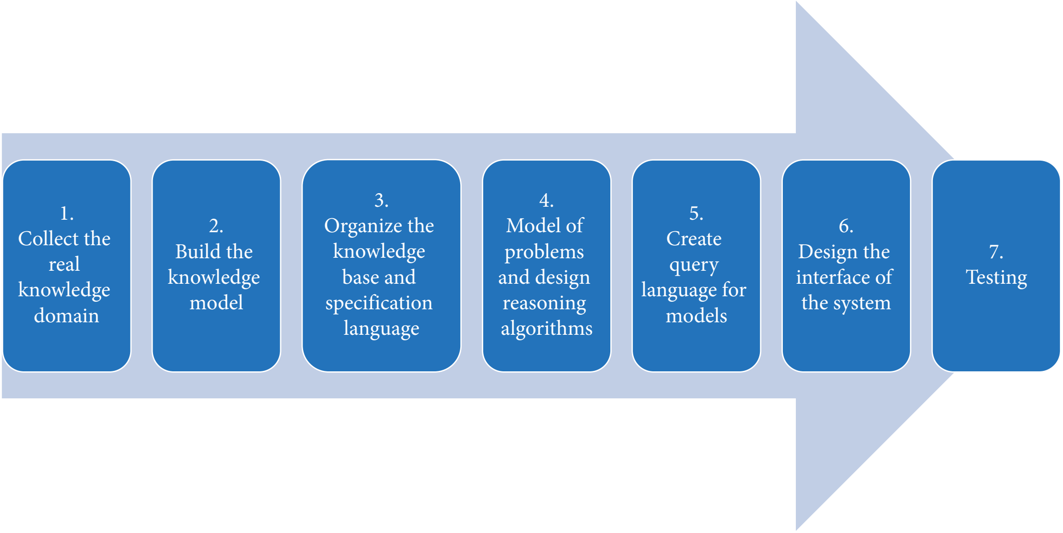
1. Collect the real knowledge domain
2. Build the knowledge model
3. Organize the knowledge base and specification language
4. Model of problems and design reasoning algorithms
5. Create query language for models
6. Design the interface of the system
7. Testing

Figure 3: The process of designing an IPS in education.

knowledge level of students. The communication of the system is pedagogical and similar to the tutoring of the lecturer. In stage 6 and stage 7, the IPS is completed by designing its interface and testing.

Stage 1. Determine the knowledge of courses and scope; then, collect the real knowledge consisting of concepts and objects, relations, operators and functions, facts, and rules. This knowledge collecting can be classified in some ways such as chapters, topics, or subjects; based on this classification, problems and exercises in the course can be collected appropriately and quickly. Problems are also classified according to some methods such as frame-based problems and general forms of problems.

This stage ensures that the knowledge domain will be represented entirely.

Stage 2. Build the model for the collected knowledge domain.

It is an essential base for designing the knowledge base of the IPS in education. The model has to represent the kernel of the knowledge domain, including concepts, relations, and rules. The kernel can be integrated with other knowledge components to represent the knowledge of the course sufficiently.

The structure of knowledge components in the model has been constructed based on the mathematical foundation. This structure is an integral part for the formality of the model.

Stage 3. Establish a knowledge base organization for the system. This stage makes the representation more natural and suitable for the knowledge level of users.

Design the specification language to represent components of the knowledge model. The knowledge engineer uses this language, which is designed to simulate the way of describing the knowledge in practice. Based on this language, a knowledge base can be organized by structured text files [5].

Stage 4. Modeling of problems and designing algorithms for automated reasoning.

Classes of problems are modeled as well to obtain initial problem models. The model of problems belongs to the structure of the knowledge model. The model of general problems usually consists of three parts: $O = \{O_1, \ldots, O_n\}$, $F = \{f_1, \ldots, f_k\}$, and

$$G = [g_1, \ldots, g_m]. \tag{1}$$

Here, set O is the set of objects, F is the set of facts given on the objects, and G is a list of goals of the problem.

Three steps for modeling can develop the design of deductive reasoning algorithms for solving problems and the design of the interface of the system:

Step 1: classify problems such as problems as frames, problems of a determination or a proof of a fact, and problems of finding objects or facts.
Step 2: classify facts in the knowledge domain.
Step 3: modeling kinds of problems from classifying in steps 1 and 2. From models of each kind, we can construct a general model for problems, which are given to the system for solving them.

The basic technique for designing deductive algorithms is the unification of facts. Based on the kinds of facts and their structures, there will be criteria for unification proposed. Then, it produces algorithms to check the unification of two facts. The next important work is researching reasoning strategies to solve problems on the computer. The most challenging thing is modeling for experience, sensible reaction, and intuitional humans to find heuristic rules, which were able to imitate human thinking for solving problems.

When designing deduction algorithms, the effectiveness and complexity of those algorithms need to be considered. Those algorithms have to be built based on the way of learners' thinking to solve problems. This stage serves the usefulness of the system to enhance studying.

Stage 5. Create a query language for the models. The query language has to be suitable for the knowledge level of students and helps to design the communication between the system and users. Inputting the problem and understanding the solution from the system is more manageable by using the query language. Moreover, the communication of the system is pedagogical and similar to the tutoring of the lecturer.

Stage 6. Design the interface of the system and coding to produce the application. Intelligent applications for solving problems in education of mathematics, physics, and chemistry have been implemented by using programming tools and computer algebra systems such as Visual Basic.NET or C#, SQL Server, and Maple [26]. They are straightforward to use for students to search, query, and solve problems.

Stage 7. Testing, maintaining, and developing the application. This stage is similar to what happens in other computer systems.

5. Knowledge Model for an Intelligent Problem Solver in Education

5.1. Rela-Ops Model

Definition 1 (see [9]). A knowledge model of relations and operators, called Rela-Ops model, is a tube:

$$(\mathrm{C}, \mathrm{R}, \mathrm{Ops}, \mathrm{Rules}) \tag{2}$$

In which:

(i) C is a set of concepts. Each concept c is a class of objects, and it has an instance set, called I_c. Each concept c is a tube *(Attrs, Facts, EqObj, RulObj)*, which *Attrs* is a set of attributes, *Facts* is a set of facts of a concept c, *EqObj* is a set of equations of a concept c, and *RulObj* is a set of deductive rules of a concept c.

(ii) R is a set of relations between concepts in C. It includes hierarchical relations and binary relations between concepts in C.

(iii) Ops is a set of operators between concepts in C. It includes unary and binary operators.

(iv) Rules is a set of inference rules of the knowledge domain. In this study, Rules-set is classified into four kinds of rules: deductive rules, rules for generating a new object, equivalent rules, and equation rules.

An inference rule $r \in$ Rules is one of the four cases:

$$\mathrm{Rules} = \mathrm{Rule}_{\mathrm{deduce}} \cup \mathrm{Rule}_{\mathrm{generate}} \cup \mathrm{Rule}_{\mathrm{equivalent}} \cup \mathrm{Rule}_{\mathrm{equation}}. \tag{3}$$

(i) $r \in \mathrm{Rule}_{\mathrm{deduce}}$: r is a deductive rule, it has the form: $u(r) \longrightarrow v(r)$ with $u(r), v(r)$ are sets of facts.

(ii) $r \in \mathrm{Rule}_{\mathrm{generate}}$: r is a rule for generating a new object, it has the form: $u(r) \longrightarrow v(r)$ with $u(r), v(r)$ satisfy: $\exists o,\ o \in v(r)$ and $r \notin u(r)$

(iii) $r \in \mathrm{Rule}_{\mathrm{equivalent}}$: r is an equivalent rule, it has the form: $h(r), u(r) \longleftrightarrow v(r)$ with $h(r), u(r), v(r)$ satisfy:

$h(r)$, $u(r) \longrightarrow v(r)$, and $h(r)$, $v(r) \longrightarrow u(r)$ are true.

(iv) $r \in \text{Rule}_{\text{equation}}$: is an equation rule, it has the form: $g(o_1, o_2, \ldots, o_k) = h(x_1, x_2, \ldots, x_p)$ with o_i, x_i are objects and g, h are expressions between objects.

The detailed structure of each component in the Rela-Ops model has been presented in [9]. This model is built based on ontology and object-oriented approaches. Each concept in the Rela-Ops model is a class of objects, and each object has the structure and behaviors to solve problems by itself.

5.2. Problems on Rela-Ops Model. In the Rela-Ops model, there are two kinds of problems: problems on an object and general problems on the model. Problems on an object are its behaviors, and they are solved based on the reasoning on their structures. General problems are solved by reasoning method on the rules in Rules-set and solving problems on objects. The solving method combines the knowledge of relations and operators to get new facts in the reasoning.

5.2.1. Problems on an Object

Definition 2 (see [9]). The closure of a set of facts.

Let Obj = (*Attrs, Facts, EqObj, RulObj*) be an object of a concept in C and F be a set of facts. The closure of set F by Obj, Obj.Closure(F), is a *maximum extension* of F by using reasoning rules in Obj.*EqObj* and Obj.*RulObj*.

There are three kinds of problems on an object in the Rela-Ops model: (1) determine the closure of a set of attributes, (2) determine the closure of a set of facts, and (3) execute deduction and give solutions for a problem. In this section, we present the algorithm to solve the problem of determining the closure of a set of facts.

Theorem 1. *The complexity of algorithm 1 is:* $O(k^{\max(n_1 \cdot q_1, n_2 \cdot q_2)} s)$

In which, $k = card(F)$*: number of facts in F.* $n_1 = card(EqObj)$*:number of equation rules in EqObj.* $q_1 = \max\{\#(g) + \#(h) | r := g = h, \ r \in EqObj\}$*.* ($\#(g), \#(h)$ *are numbers of objects in g and h, resp.).* $n_2 = card(RulObj)$*: number of deductive rules in RulObj.* $q_2 = max\{card(u(r)) \ |r \in RulObj\}$

Proof of Theorem 1. The complexity of algorithm 1 depends on the complexity of step 3, step 4, and step 5. We have

(i) The complexity of deducing objects in step 3 is $O(k^3)$

(ii) The complexity of searching rules in step 4 is $O(k^{n_1 \cdot q_1})$

(iii) The complexity of searching rules in step 5 is $O(k^{n_2 \cdot q_2})$ □

Hence, the complexity of algorithm 1 is

$$O\left(\max\left(k^3, k^{n_1 \cdot q_1}, k^{n_2 \cdot q_2}\right)\right) = O\left(\max\left(k^{n_1 \cdot q_1}, k^{n_2 \cdot q_2}\right)\right) = O\left(k^{\max(n_1 \cdot q_1, n_2 \cdot q_2)}\right). \quad (4)$$

5.2.2. General Problems on Rela-Ops Model

Definition 3. Models of problems on Rela-Ops model:

(a) *Kind 1*: model of problems has the form $(\mathbf{O}, \mathbf{F}) \longrightarrow \mathbf{G}$

where $\mathbf{O} = \{O_1, O_2, \ldots, O_m\}$ is a set of objects in problem. $\mathbf{F} = \{f_1, f_2, \ldots, f_n\}$ is a set of facts. $\mathbf{G} = \{$“KEYWORD”: $f\}$ with “KEYWORD” is a keyword of the goal and f is a sentence, “KEYWORD” may be the following:

(i) “Determine”: it means to determine a sentence f.

(ii) “Prove”: it means to prove a sentence f.

(iii) “Compute”: it means to determine the value of f when f is an expression.

(b) *Kind 2*: model of problems has the form $(\mathbf{O}, \mathbf{E}, \mathbf{F}) \longrightarrow \mathbf{G}$

where $\mathbf{E} = \{\text{expr}_1, \text{expr}_2, \ldots, \text{expr}_{\text{p}}\}$ is the set of expressions between objects in O. $\mathbf{G} = \{$ “KEYWORD”: f $\}$ with “KEYWORD” may be the following:

(i) “Reduce”: it means to reduce a sentence f when f is an expression.

(ii) “Transform”: it means to transform an object f into an expression between particular objects.

Problems in kind 1 and kind 2 were studied and solved in [6, 9, 27]. The effectiveness of the algorithms for solving problems in kind 1 has been proven in [6, 9] and for solving problems in kind 2 has been proven in [9, 27].

Lemma 1 (see [27]). Let a knowledge domain K as Rela-Ops model and $(\mathbf{O}, \mathbf{E}, \mathbf{F})$ be the hypothesis of the problem as kind 2 in Definition 3. There exists a unique maximum set $L_{(O,\ E,\ F)}$ such that it contains all facts that can be deduced from $(\mathbf{O}, \mathbf{E}, \mathbf{F})$.

Theorem 2 *(see [27]). Let a knowledge domain K as Rela-Ops model and a problem* $P = (\mathbf{O}, \mathbf{E}, \mathbf{F}) \longrightarrow \mathbf{G}$ *as kind 2 in Definition 3. Suppose* $S = [s_1, s_2, \ldots, s_k]$ *is a list of rules. The following statements are equivalent:*

(i) Problem P is solvable

(ii) $\mathbf{G}\ .f \in L_{(O,\ E,\ F)}$

(iii) There exists a list of rules $S = [s_1, s_2, \ldots, s_k]$ *such that* $\mathbf{G}.f \in S(\mathbf{E}, \mathbf{F})$*, with* $S(\mathbf{E}, \mathbf{F})$ *is a set of facts can be deduced from the list S and hypothesis of problem P*

Theorem 2 shows that forward chaining reasoning will deduce the goals of problems. Besides, algorithm 2 is designed based on forward chaining; therefore, Theorem 2 guarantees the effectiveness of this algorithm.

Let $K = (\mathbb{C}, \mathbb{R}, \mathbb{O}ps, \mathbb{R}ules)$ be a knowledge domain as Rela-Ops model, Obj = (*Attrs, Facts, EqObj, RulObj*) be an object of a concept in C, F is a set of facts. This algorithm deduces the closure of set F by Obj, Obj.Closure(F).

Input: Object Obj = (*Attrs, Facts, EqObj, RulObj*), F is a set of facts.

Output: Obj.Closure(F)

```
Step 0: Initialize variables
    flag:= true;
    KnownFacts:= $F \cup$ Obj.Facts;
Step 1. Classify kind of facts in KnownFacts
Step 2. Determine new facts from facts in KnownFacts by using reasoning rules.
Step 3. Search the closure of facts as an object in KnownFacts.
    for fact in KnownFacts do
        if (fact is an object) then
        KnownFacts:= KnownFacts $\cup$ fact.Attrs;
        end if;
    end do;
Step 4. Search the rule in Obj.EqObj which can be applied based on KnownFacts.
flag:= true;
while (flag! = false) do
    4.1. if (a rule r in Obj.EqObj can be found) then
        r has form: $g(x_1, \ldots, x_n) = h(y_1, \ldots, y_m)$ which g, h are epressions, $x_i$, $y_j \subseteq$ Obj.Attrs $(1 \leq i \leq n,\ 1 \leq j \leq m)$
        Combine facts in Knowfacts for solving equation $g(x_1, \ldots, x_n) = h(y_1, \ldots, y_m)$ to determine new attributes.
        Update KnownFacts.
    end if; # 4.1
    4.2. if (cannot be found a rule $r \in$ Obj.EqObj) then
        flag:= false;
    end if;
end do;
Step 5. Search the rule in Obj.RulObj which can be applied based on KnownFacts
while (flag! = false) do
    5.1. if (a rule r in Obj.RulObj can be found) then
        r has form: $u(r) \longrightarrow v(r)$ which $u(r) \subseteq$ Obj.Attrs and $v(r) \subseteq$ Obj.Attrs
        for e in $v(r)$ do
            KnownFacts:= KnownFacts $\cup \{e\}$;
            if (new facts can be determined from KnownFacts) then
                Determine new facts from facts in KnownFacts by using deduce rules;
            end if;
            if (e is a new object) then
                KnownFacts:= KnownFacts $\cup$ e.Closure($v(r)$);
                if (new facts can be determined from KnownFacts) then
                    Determine new facts from facts in KnownFacts by using deduce rules;
                end if;
            end if;
        end do;
    end if; # 5.1
    5.2. if (cannot be found a rule $r \in$ Obj.RulObj) then
    flag:= false;
    end if;
end do; # while
Step 6. Obj.Closure(F):= KnownFacts
```

Algorithm 1: Determine the closure of a set of facts.

5.2.3. Rela-Ops Model and Criteria of a Knowledge Model for an IPS in Education. Rela-Ops model is a knowledge model, including the knowledge of relations and operators. These kinds of knowledge are popular in practice, especially in STEM knowledge. This model is flexible and effective in practical applications. As shown in the appendix

Let $K = (\mathbb{C}, \mathbb{R}, \mathbb{O}\text{ps}, \mathbb{R}\text{ules})$ be a knowledge domain as Rela-Ops model, and a problem $P = (\mathbf{O}, \mathbf{E}, \mathbf{F}) \longrightarrow \mathbf{G}$ as kind 2 in Definition 3. This algorithm will solve problem P through these steps as follows:

Input: The problem $P = (\mathbf{O}, \mathbf{E}, \mathbf{F}) \longrightarrow \mathbf{G}$

Output: The solution to problem P.

The method for designing this algorithm uses forward chaining reasoning. It combines heuristics rules in the reasoning process. Objects also attend this process as active agents for solving problems on themselves by Algorithm 1. This process is done when it gets the goal.

```
 Step 0: Initialize variables
   flag:= true;
   KnownFacts:= $\mathbf{E} \cup \mathbf{F}$;
   count:= 0; # the number of new objects which are generated
   Sol:= []; # solution of problem
 Step 1. Collect objects in hypothesis and goal part.
   Classify kind of facts in E and F.
 Step 2. Check G.
   If G is achieved then
     Go to step 5.
 Step 3: Determine the closure of each object in O by using Algo. 4.1 and facts in E and F.
 Step 4: Use equations in E to generate the new facts as relation form.
   Use the relations in F to generate new equations.
   Update KnownFacts.
 Step 5: Select a rule in Rules-set to produce new facts or new objects by using heuristic rules.
 while (flag! = false) and not(G is determined) do
   Search r in Rules which can be applied to KnownFacts
   5.1. Case: r is a deductive rule
     if (r has form: $h(r) \longrightarrow g(r)$) then
     KnownFacts:= $KnownFacts \cup g(r)$;
     s:= [r, h(r), g(r)];
     Sol:= [op(Sol), s];
     continue;
     end if;
   5.2. Case: r is a rule for generating a new object
     if $count \le \text{card}(O)$ then #only generate at most number of objects in hypothesis
 if (r generates a new object o) and not($o \in KnownFacts$)
   then
     count:= count + 1;
     KnownFacts:= $KnownFacts \cup g(r)$; s: = [r, h(r), g(r)];
     Sol:= [op(Sol), s];
     Go to Step 3 with new object o;
     end if;
   end if; #5.2
   5.3. Case: r is an equivalent rule
     if (r has form: f(r), $h(r) \leftrightarrow g(r)$) then
     KnownFacts:= $KnownFacts \cup g(r)$;
   s: = [r, h(r), g(r)];
     Sol: = [op(Sol), s];
     continue;
 end if; #5.3
   5.4. Case: r is an equation rule
   if (r has form: $u = v$) then
   r can generate a set of new facts A
     KnownFacts:= $KnownFacts \cup A$;
   s:= [r, KnownFacts, A];
     Sol:= [op(Sol), s];
 if (r generates a new object o) and not($o \in KnownFacts$) then
     count:= count + 1;
     Go to Step 3 with new object o;
   end if;
 end if; #5.4
   5.5. if (rule r cannot be found) then
     flag:= false;
```

Algorithm 2: Continued.

```
        end if;
    end do ; #while
    Step 6: Conclusion of the problem
        if G is determined then
            Problem (O, E, F) ⟶ G is solvable;
            Sol is a solution to the problem;
    Reduce Sol by eliminating redundant rules.
        else
            Problem (O, E, F) ⟶ G is unsolvable;
            end if;
```

Algorithm 2: (see [27]). Solving the problem as kind 2.

(Supplementary Materials (available here)), this model can apply to design knowledge bases of courses about vector algebra in the high-school mathematics, direct current (DC) electrical circuit in the middle-school physics, and programming course about data structure and algorithms in the university. Some models of reducing the Rela-Ops model has been used to represent knowledge domains: the model of knowledge of relations (C, R, Rules) for solid geometry in high-school [6, 28] and the model of knowledge of operators (C, Ops, Rules) for discrete mathematics in university [8]. These knowledge bases can be applied in corresponding IPS systems. Those representations by the Rela-Ops model are naturalness. The input and output of exercises in these courses are easy to use and understand. Solutions of them are step-by-step, and their reasoning is like the solving method of students.

The detailed structure of the Rela-Ops model and its problems has been presented in [6, 27]. The finiteness, the effectiveness, and the complexity of algorithms have also been proved in [6, 9, 27].

6. Discussion

Nowadays, knowledge representation methods can be classified into four types: the representation by formal logic, networks, ontology, and algebraic approach.

Formal logic methods are not effective for the complex knowledge domains, especially in education. Besides classical logic methods, description logic has also been studied. This logic is the formal representation of semantic [22, 23]. However, logic methods cannot represent STEM knowledge, mainly structural and relational knowledge. Hence, they cannot be applied to the design of the knowledge base of an IPS in STEM education.

Representation methods by networks are suitable for classifying the concepts. These methods are not valid for the practical knowledge domain, especially computing knowledge. A semantic network belongs to the language for representation. A *knowledge graph* is a methodology to perform link prediction between entities. Its nodes represent the item, entity, and user, and its edges represent the linking nodes that interact with each other [29]. The knowledge graph is a useful tool for information searching and giving semantics to textual information. Nonetheless, it is difficult to reason for solving problems, especially the problems of an IPS system in STEM education.

Algebraic approach is a representation method based on the mathematical structures; they are classical algebraic structures, such as groups, rings, ideals, and fields, or they are integrating those structures [24]. The problem of information equivalence of knowledge has been solved in [30] based on the definition of the symmetries of knowledge bases. The knowledge base as logic is also presented by the structure of matrices in linear algebra [31]. The knowledge in these results only has information form; hence, they cannot be applied to solve significant problems that require the ability to reason in the problem-solving process.

In intelligent tutoring systems, *ontology* is used as a framework to represent the content of a course [32]. These systems could not yet solve problems automatically. Computational Network Object Knowledge Base (COKB) is an ontology that can be applied to build practical applications in IPS systems [5]. However, the formality of this model has some limitations. The mathematical foundation of COKB's components has not yet been presented clearly.

Rela-Ops model can satisfy the criteria of a knowledge representation method for an IPS in education, especially for technological courses, such as mathematics, physics, and chemistry. It can represent many kinds of knowledge domains in education, such as mathematics, physics, and programming. The IPS systems built based on it are useful for students. They can solve common exercises in corresponding courses and some hard problems with them. Their solutions are step-by-step. Their reasoning is appropriate to the knowledge level of learners. In practice, some knowledge domains include many subdomains; thus, for representing those knowledge domains, the representation method has to support the integrating of knowledge bases between subdomains. The architecture of the Rela-Ops model can integrate subdomains which have the structure as the Rela-Ops model. The integration between knowledge bases for designing an IPS has been studied in [4]. For example, Appendix C (Supplementary Materials (available here)) presents an integrating model between Rela-Ops model and frames [33]. This model is used to represent the knowledge base of programming and design the intelligent system for learning of courses about algorithms [33, 34].

Table 6: Comparison between methods for knowledge representation.

Method	Generality	Usability	Practicality	Formality
Formal logic	Level 2	Level 1	Level 1	Level 4
Networks	Level 2	Level 3	Level 2	Level 3
Ontology	Level 3	Level 3	Level 2	Level 3
Algebraic approach	Level 1	Level 2	Level 1	Level 4
Rela-Ops model	Level 4	Level 4	Level 3	Level 3

Table 6 compares the discussed methods for knowledge representation, as far as the satisfaction of criteria of knowledge models for IPS systems are concerned.

7. Conclusions and Future Work

In this paper, the criteria of a knowledge model for an intelligent problem solver in STEM education have been proposed. They include generality, usability, naturalness, and formality. Each criterion has certain levels. These criteria orient to develop a method for knowledge representation about theory and application. The knowledge base, which is built based on those criteria, can meet the requirements of an IPS.

(i) The *generality* criterion executes the compatibility of a knowledge representation method for different knowledge domains of courses.

(ii) This *usability* criterion is the completeness criterion for intelligent software requirements specification standards.

(iii) The *naturalness* criterion is the practicality criterion of software. It is the nature of the IPS works. The representation method has to guarantee the nature of specification language and the method of reasoning to accomplish the naturalness criterion.

(iv) The *formality* criterion ensures the correctness of the representation method. This criterion supports theoretical evidence for the effectiveness of the method.

For proving the effectiveness of these criteria, the Rela-Ops model is introduced in this paper. It is a model representing the combining knowledge of relations and operators. This model is built based on the object-oriented and ontological approach. Each concept in Rela-model is a class of objects which also have the structure and the ability to solve problems on themselves. Rela-model can be applied to design the knowledge bases of IPS systems for corresponding courses. It also satisfies the criteria of a knowledge model for an IPS in STEM education.

The real-word knowledge domain has many sub-domains, so the criteria of knowledge representation method have to mention to the problems about integrating knowledge-based systems. In the future, we will continue to study these criteria of an integrated knowledge model for an IPS. From that, they will be developed to be the criteria of a general knowledge model. Those results will be the foundation for building a supporting tool to design general knowledge-based systems. Besides, the integration method of knowledge bases, which are as Rela-Ops, needs to study for application in IPS.

Data Availability

No data were used to support this study.

Disclosure

This paper is a revised and expanded version of a paper entitled Criteria of a Knowledge Model for an Intelligent Problem Solver in Education**,** Proceedings of 10th IEEE International Conference on Knowledge and Systems Engineering (KSE 2018), pp. 288-293, Ho Chi Minh City, Vietnam, Nov. 2018.

Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

Acknowledgments

This research was funded by Vietnam National University HoChiMinh City (VNU-HCM) under grant no. C2019-26-01.

Supplementary Materials

Appendix A: intelligent problem solver in vector algebra. Appendix B: intelligent problem solver in direct current electrical circuits. Appendix C: intelligent system for learning data structure and algorithms. (*Supplementary Materials*)

References

[1] Y. Xie, M. Fang, and K. Shauman, “STEM education,” *Annual Review of Sociology*, vol. 41, no. 1, pp. 331–357, 2015.

[2] Q. M. Clark and J. V. Clark, “STEMulate-K12: automating STEM attraction,” in *Proceedings of 2018 IEEE Frontiers in Education Conference (FIE)*, San Jose, CA, USA, October 2018.

[3] R. Murphy, *Artificial Intelligence Applications to Support K–12 Teachers and Teaching: A Review of Promising Applications, Challenges, and Risks*, RAND, Santa Monica, CA, USA, 2019.

[4] N. V. Do, H. D. Nguyen, and T. T. Mai, “A method of ontology integration for designing intelligent problem solvers,” *Applied Sciences*, vol. 9, no. 18, p. 3793, 2019.

[5] N. V. Do, “Intelligent problem solvers in education: design method and applications,” in *Intelligent Systems, Chapter 6*, V. M. Koleshko, Ed., IntechOpen, London, UK, pp. 121–148, 2012.

[6] N. V. Do, H. D. Nguyen, and A. Selamat, “Knowledge-based model of expert systems using rela-model,” *International Journal of Software Engineering and Knowledge Engineering*, vol. 28, no. 8, pp. 1047–1090, 2018.

[7] H. D. Nguyen, N. V. Do, N. P. Tran, and X. H. Pham, “Criteria of a knowledge model for an intelligent problems solver in education,” in *Proceedings of 2018 10th IEEE International Conference on Knowledge and Systems Engineering (KSE*

2018), pp. 288–293, Ho Chi Minh City, Vietnam, November 2018.

[8] H. D. Nguyen and N. V. Do, "Intelligent problems solver in education for discrete mathematics," in *Proceedings of 16th International Conference on Intelligent Software Methodologies, Tools, and Techniques (SOMET_17)*, vol. 297, pp. 21–34Frontiers in Artificial Intelligence and Applications, Kitakyushu, Japan, September 2017.

[9] H. D. Nguyen, N. V. Do, and V. T. Pham, "Rela-Ops model: a method for knowledge representation and application," in *Proceedings of 17th international conference on intelligent software methodologies, tools, and techniques (Somet 2018)*, vol. 303, pp. 825–838Frontiers in Artificial Intelligence and Applications, Granada, Spain, September 2018.

[10] R. Davis, H. Shrobe, and P. Szolovits, "What is a knowledge representation?" *AI Magazine*, vol. 14, no. 1, pp. 17–33, 1993.

[11] N. Noy, "Final report on the 2013 NSF workshop on research challenges and opportunities in knowledge representation," National Science Foundation Workshop Report, Wright State University, Dayton, OH, USA, 2013.

[12] K. AbuDahab, D.-l. Xu, and Y.-w. Chen, "A new belief rule base knowledge representation scheme and inference methodology using the evidential reasoning rule for evidence combination," *Expert Systems with Applications*, vol. 51, pp. 218–230, 2016.

[13] R. J. Brachman and H. J. Levesque, *Knowledge Representation and Reasoning*, Elsevier, Amsterdam, Netherlands, 2004.

[14] Y. Wang and M. Valipour, "Formal ontology generation by deep machine learning," in *Proceedings of 2017 IEEE 16th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)*, Oxford, UK, July 2017.

[15] C. Sakama, K. Inoue, and T. Sato, "Linear algebraic characterization of logic programs," in *Proceedings of 10th International conference on Knowledge Science, Engineering and Management (KSEM 2017), LNAI 10412*, Springer, Melbourne, Australia, pp. 520–533, August 2017.

[16] E. Siahlooei and S. Fazeli, "Two iterative methods for solving linear interval systems," *Applied Computational Intelligence and Soft Computing*, vol. 2018, Article ID 2797038, 13 pages, 2018.

[17] H. Salim, K. Faisal, and R. Jawad, "Enhancement of performance for steam turbine in thermal power plants using artificial neural network and electric circuit design," *Applied Computational Intelligence and Soft Computing*, vol. 2018, Article ID 8042498, 9 pages, 2018.

[18] R. Nkambou, J. Bourdeau, and V. Psyché, "Building intelligent tutoring systems: overview," in *Advances in Intelligent Tutoring Systems*, R. Nkambou, Ed., pp. 361–375, Springer, Berlin, Germany, 2010.

[19] I. Hatzilygeroudis and J. Prentzas, "Knowledge representation requirements for intelligent tutoring systems," in *Proceedings of 7th International Conference on Intelligent Tutoring Systems (ITS 2004), LNCS 3220*, pp. 87–97, Springer, Maceió, Brazil, August 2004.

[20] I. Hatzilygeroudis and J. Prentzas, "Chapter 9: knowledge representation in intelligent educational systems," in *Web-Based Intelligent E-Learning Systems: Technologies and Applications*, Z. Ma, Ed., IGI Publisher, Pennsylvania, PA, USA, 2006.

[21] E. A. Giakoumakis and G. Xylomenos, "Evaluation and Selection criteria for software requirements specification standards," *Software Engineering Journal*, vol. 11, no. 5, pp. 307–319, 1996.

[22] A. Y. Turhan, "Introductions to description logics - a Guided tour," in *Proceedings of 9th International Summer School on Reasoning Web Semantic Technologies for Intelligent Data Access, LNCS 8067*, Springer, Mannheim, Germany, pp. 150–161, July 2013.

[23] C. Sakama, K. Inoue, and T. Riberio, "Learning inference rules from data," *Künstliche Intelligenz (KI)*, vol. 33, pp. 267–278, 2019.

[24] J. Calmet and I. A. Tjandra, "Building bridges between knowledge representation and algebraic specification," *Lecture Notes in Computer Science, LNCS*, vol. 869, pp. 295–304, 1994.

[25] N. V. Do, H. D. Nguyen, and T. T. Mai, "Designing an intelligent problems solving system based on knowledge about sample problems," in *Proceedings of 5th Asian conference on Intelligent Information and Database Systems (ACIIDS 2013)* Springer, Kuala Lumpur, Malaysia, March 2013.

[26] Maple, https://www.maplesoft.com/.

[27] H. D. Nguyen, N. V. Do, V. T. Pham, and K. Inoue, "Solving problems on a knowledge model of operators and application," *International Journal of Digital Enterprise Technology (IJDET)*, vol. 1, no. 1/2, pp. 37–59, 2018.

[28] H. D. Nguyen, D. Nguyen, and V. T. Pham, "Design and intelligent problems solver about solid geometry based on knowledge model about relation," in *Proceedings of 2016 IEEE International Conference on Knowledge and Systems Engineering (KSE 2016)*, pp. 150–155, Ha Noi, Vietnam, October 2016.

[29] Y. Duan, L. Shao, G. Hu, and Z. Zhou, "Specifying architecture of knowledge graph with data graph, information graph, knowledge graph and wisdom graph," in *Proceedings of SERA 2017*, pp. 327–332, London, UK, June 2017.

[30] T. Plotkin and M. Knyazhansky, "Symmetries of knowledge bases," *Annals of Mathematics and Artificial Intelligence*, vol. 64, no. 4, pp. 369–383, 2012.

[31] C. Sakama, H. D. Nguyen, T. Sato, and K. Inoue, "Partial evaluation of logic programs in vector space," in *Proceedings of the 11th Workshop on Answer Set Programming and Other Computing Paradigms (ASPOCP 2018), co-located with Federated Logic Conference 2018 (FLoC 2018)*, Oxford, UK, July 2018.

[32] H. Wang, "Research on the model of knowledge representation ontology based on framework in intelligent learning system," in *Proceedings of 2011 International Conference on Electrical and Control Engineering (ICECE 2011)*, pp. 6757–6760, Yichang, China, September 2011.

[33] T. T. Le, S. T. Luu, H. D. Nguyen, and N. V. Do, "Knowledge representation method for designing an intelligent tutoring system in learning of courses about algorithms," in *Proceedings of 2019 25th Asia-Pacific Conference on Communications (APCC 2019)*, pp. 310–315, Ho Chi Minh, Vietnam, November 2019.

[34] H. D. Nguyen, N. V. Do, T. T. Mai, and V. T. Pham, "A method for designing the Intelligent system in learning of algorithms," in *Proceedings of 18th International Conference on Intelligent Software Methodologies, Tools, and Techniques (SOMET 2019)*, vol. 318, pp. 658–671Frontiers in Artificial Intelligence and Applications, Kuching, Malaysia, September 2019.

[35] H. V. Tran, M. H. Nguyen, D. V. Nguyen, and D. H. Tran, *Textbook and Workbook of Geometry at High School*, Publisher of Education, Ho Chi Minh City, Vietnam, 4th edition, 2014.

[36] Q. VuD. H. Doan et al., *Textbook and Workbook of Physic at Middle School*, Publisher of Education, Ho Chi Minh City, Vietnam, 9th edition, 2014.

[37] V. N. Do and H. D. Nguyen, "Reducing model of COKB about operators knowledge and solving problems about operators,"

in *New Trends in Computational Collective Intelligence*, D. Camacho, Ed., pp. 39–49, Springer, Berlin, Germany, 2015.

[38] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, Cambridge, MA, USA, 3rd edition, 2009.