A Reasoning Method on Knowledge Base of Computational Objects and Designing a System for Automatically Solving Plane Geometry Problems

Nhon Do, Hien Nguyen

***Abstract*— In artificial intelligence, there are many methods for knowledge representation. Nonetheless, these methods are not efficient for executing and reasoning on complex knowledge. Ontology, a modern approach, has been studying deeply due to its ability to represent knowledge. The Knowledge Base of Computational Objects (KBCO) model is an ontology which can be used efficiently for designing complex knowledge base systems in the areas: Geometry, Analysis, Physics, etc. Besides, inference methods also play an important role in knowledge base systems. However, the current methods are too general to imitate human's way of thinking. In reality, when solving a problem, we often start by finding the problems which, in some sense, are relevant to the current problem. In this paper, we present an extended model of the KBCO model. Sample problems will be used as available knowledge in a way that imitate more optimally human's knowledge. KBCO model using Sample Problems can be applied to construct complex intelligent systems to simulate some knowledge domains of human.**

***Keywords*—artificial intelligence, educational software, reasoning method, knowledge representation.**

I. Introduction

Models and methods for knowledge representation play an important role in designing knowledge base systems and expert systems. Nowadays there are many various knowledge models, such as semantic network, neural network, conceptual graph, etc (see [1,2,3,4]). Nonetheless, these models only can represent an aspect of practical knowledge. A domain of real knowledge is generally diverse and contains simple-to-complex concepts. In [21], another model for knowledge representation, called Computation Network, has been proposed and utilised for practical applications. Furthermore, a modern approach for knowledge representation and knowledge base system designing is using "ontology" (see [6, 9, 20]). In particular, the Knowledge Base of Computatiational Objects (KBCO) model proposed in [14, 16, 22] is very useful and suitable for representing knowledge in the complex domains of practical applications, such as Analytical Geometry, Plane Geometry, Physics, Chemistry, etc.

For automatic inference methods, the most important thing is controlling strategy for generating new facts from known facts. Many automatic inference techniques have been studied quite completely in a general level, including: (see [5, 7, 8, 10, 12])

1. Unification routine in methodology for knowledge base representation by first-order logic.
2. Inference by forward chaining
3. Inference by backward chaining
4. Combining forward and backward chaining together with using heuristic rules.

However, the above results are still too general. Some models and techniques are still partial and not good enough for constructing a knowledge system with requirements which are not easy for designer to set up and implement.

In order to deal with a practical problem, we often cannot immediately find a new solution. In [11], the author pointed out that we first need to answer the following questions:

" - *Have you seen it before ? - Can you think of a related problem ?- Look at the unknown and try to recall a more familiar problem that has a similar unknown. - Here is a problem related to yours and solved before. Could you use it ? Could you use its result? Could you use its method?* "

(G. Polya)

This corresponds to searching relating problems which were solved before, and then proposing an appropriate solution for the problem. Besides, we can use the result of relating problems for solving the current problem. Such relating problems are called *Sample Problems*.

Analogical reasoning which is a method to deduce based on sample problems has been metioned in [23]. However, this method is still general and very difficult to apply in complex real knowledge.

In this paper, we extend KBCO model by adding sample problems component to the knowledge of the system. This extension allows us to imitate human's thought about finding sample problems via answering the following questions:

(1) Have I solved this problem before? Or have I solved some kinds of this problem before ?
(2) This is a problem I solved before. Can I apply its result?

This extension model of KBCO is necessary for representing knowledge base. In addition, this extension model can be applied for modeling knowledge domain about plane geometry in middle school and constructing a system for automatic solving problems on this knowledge domain.

Nhon Do is currently a senior lecturer in the faculty of Computer Science at the University of Information Technology, Ho Chi Minh City, Vietnam. He got his MSc and Ph.D. in 1996 and 2002 respectively, from University of Sciences – National University of Ho Chi Minh City. His research interests include Artificial Intelligence, computer science, and their practical applications, especially intelligent systems and knowledge base systems. (email: nhondv@uit.edu.vn)

Hien Nguyen is a master student at University of Sciences – National University of Ho Chi Minh City, Vietnam. His research interests include computer science, Artificial Intelligence, especially knowledge representation. He is currently a teaching assistant in the faculty of Computer Science at the University of Information Technology, Ho Chi Minh City, Vietnam. (email: hiennd@uit.edu.vn)

ISBN: 978-988-18210-9-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

II. MODEL FOR KNOWLEDGE BASE OF COMPUTATIONAL OBJECTS

The model for knowledge base of computational objects has been presented in [14]. In this section, we summary some main ideas of this model.

A. Component of KBCO model:

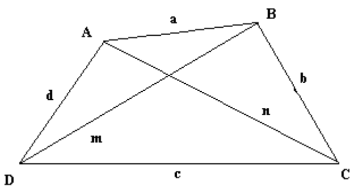
Definition 2.1: The model for knowledge base of computational objects (KBCO model) consists of 6 components:

$$\mathcal{K} = \mathbf{(C, H, R, Ops, Funcs, Rules)}$$

The meanings of the components are:

- **C** is a set of concepts of computational objects (Com-Object).
- **H** is a set of hierarchy relation on the concepts.
- **R** is a set of relations on the concepts.
- **Ops** is a set of operators.
- **Funcs** is a set of functions.
- **Rules** is a set of rules.

Each concept in C is a class of Com-objects. The structure Com-Objects can be modeled by **(*Attrs, Relas, Facts, RulesObj*)**. *Attrs* is a set of attributes, *Relas* is a set of equations called computation relations, *Facts* is a set of properties or events of objects, and *RulesObj* is a set of deductive rules on facts. For example, knowledge about a Quadrangle consists of elements (angles, edges, etc) together with formulas and some properties on them can be modeled as a class of Com-objects whose sets are as follows:



Attrs = {A, B, C, D, a, b, c, d, m, n, S, p, ...} is the set of all attributes of a Quadrangle,

Relas = {A + B + C + D = π, p = (a + b + c + d) / 2,
n = a^2+b^2-2.a.b.cos(B), etc.}

Facts = {a+b>n; a+d>m; b+c>m ; etc.}

RulesObj = { {m $\perp$ n} $\Rightarrow$ {S = ½.m.n}, etc. }

B. Kind of facts in KBCO model:

In the KBCO model there are 11 kinds of facts accepted. These kinds are as follow:

- **<u>Fact of kind 1</u>**: information about object kind.
  E.g.: [d,"LINE"] ; [O, "TRIANGLE"] ; [M,"POINT"]
- **<u>Fact of kind 2</u>**: a determination of an object or an attribute of an object.
  E.g.: SEGMENT[A,B] TRIANGLE[A,B,C].area
- **<u>Fact of kind 3</u>**: a determination of an object or an attribute of an object by a value or a constant expression.
  E.g.: ANGEL[D,E,F] = $\pi / 3$
  TRIANGLE[A,B,C].area = 12 cm^2
- **<u>Fact of kind 4</u>**: equality on objects or attributes of objects.
  E.g.: SEGMENT[A,B] = SEGMENT[M,N]
- **<u>Fact of kind 5</u>**: a dependence of an object or an attribute of an object on other objects by a formula.
  E.g.:
  ANGEL[A,B,C]=ANGEL[A,B,D]+ANGEL[D,B,C]
- **<u>Fact of kind 6</u>**: a relation on objects or attributes of objects.
  E.g.: ["PERDENTICULAR",d1,d2]
  ["BELONG",M,d]
- **<u>Fact of kind 7</u>**: a determination of function
- **<u>Fact of kind 8</u>**: a determination of a function by a value or a constant expression.
- **<u>Fact of kind 9</u>**: equality between an object or an attribute and a function.
  E.g.: M=INTESECTION(d1,d2)
  N=MIDPOINT(A,B)
- **<u>Fact of kind 10</u>**: equality between a function and another function.
  E.g: MIDPOINT(A,B) = PROJECTION(M,d)
- **<u>Fact of kind 11</u>**: a dependence of a function on other functions or other objects by an equation.
  E.g:
  DISTANCE(A,d) = SEGMENT[A,B] –
  DISTANCE(B,d)

C. Model of Problems on KBCO model:

Base on Com-objects network has been mentioned in [16] and [20], definition for modeling of problems on KBCO as followed:

*<u>Definition 2.2</u>***:** Problems are represented using a model that is called *networks of Com-Objects*. It consists of three sets below.

$O = \{O_1, O_2, \ldots, O_n\}$,
$F = \{f_1, f_2, \ldots, f_m\}$,
$G = \{ g_1, g_2, \ldots, g_k \}$.

In the above model the set **O** consists of n Com-objects, **F** is the set of facts given on the objects, **(O, F)** is a network of Com-objects, and **G** consists of goals. A goal of a problem may be the followings:

- Determine an object.
- Determine an attribute (or some attributes) of an object.
- Consider a relation between objects.
- Compute a parameter (or some parameters).
- Compute a value of a function relative to objects.

We consider the problem that to determine (or compute) attributes in set **G** from given attributes in set L = O $\cup$ F. The problem will be denoted by **L → G**

III. KBCO MODEL WITH SAMPLE PROBLEMS

When dealing with a practical problem, a convenient way to proceed is considering whether we have met a similar or related problem before or not. If so, then the solution for the problem can be obtained effectively. Or we determine whether the result of relating problems can be used to solve the practical problem or not. This leads to a requirement that model of knowledge base needs to be added a new component which can capture this behavior of human. With this adding, the inference engine can find and use results already known quickly.

ISBN: 978-988-18210-9-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

A. Knowledge sub-domain of knowledge which was presented by KBCO model:

Definition 3.1: Give knowledge domain $\mathcal{K}$ = (C, H, R, Ops, Funcs, Rules) as definition 2.1, knowledge sub-domain of $\mathcal{K}$ is a knowledge domain which was represented by KBCO model, it consists of components as followed:

$\mathcal{K}$ ' = (C', H', R', Ops', Funcs', Rules')

where,

- **C'** $\subset$ C
- **H'** $\subset$ H
- **R'** $\subset$ R
- **Ops'** $\subset$ Ops
- **Funcs'** $\subset$ Funcs
- **Rules'** $\subset$ Rules

Components of $\mathcal{K}$ ' are defined as KBCO model. However, these components include some certain objects and facts. In the other hand, knowledge domain $\mathcal{K}$ ' is a restriction of knowledge $\mathcal{K}$.

B. Sample Problem:

Model of Sample Problem:

Definition 3.2: Give knowledge sub-domain $\mathcal{K}$ ', Sample Problem (SP) is a problem which was represented by networks of Com-Objects on knowledge $\mathcal{K}$ ', it consists of three components as followed:

(O', F', Goal')

In which, O' and F' contain objects and facts were specificated on knowledge $\mathcal{K}$ '.

Criteria of Sample Problem:

When reasoning on knowledge base for finding solution of a practical problem, we usually search for relating problems which were solved before and then apply it to find the solution of the current problem. Hence, Sample Problem is a factor that makes inference become faster and more effectively. Therefore, it is required to set criteria for determining a problem is Sampe Problem.

Base on real domains of knowledge which have been researched, such as knowledge about Geometry (2D-analytical, 3D-analytical and plane geometry), Algebra, Physics (one-way electric, circuit electric), there are three criteria determining if a problem S is Sample Problem:

- First criterion: Frequency of using problem S in the knowledge domain is high. (In the sample space of common problems based on the knowlege domain, frequency of using problem S is greater than some δ)
- Second criterion: Number of Objects in problem S is small (smaller than some λ, it means card(O_p) $\leq \lambda$)
- Third criterion: Solution of problem S is short (less than some ψ steps).

In which, (δ, λ, ψ) are constants depending on the knowledge domain. For example, in the knowledge domain about plane geometry, δ= 0.5 , λ=5 and ψ=5.

Kinds of Sample Problems:

There are two kinds of SP including:

- ***Problems about determining an Object.***

Example 1: Problems about determining a plane in analytical geometry.

Problem P1: " Give three pionts A, B, C have coordinates: A(a1,a2,a3) B(b1,b2,b3) C(c1,c2,c3)
Determine plane (Q) though three points A,B,C."

\+ Specification of Sample Problem P1:

P1 = L' → G'

In which: L' = O' ⋃ F'

O' = { [A,"POINT"], [B,"POINT"],[C,"POINT"]
[Q, "PLANE"]
}

F' = {A = [a1,a2,a3] , B = [b1,b2,b3],
C = [c1,c2,c3],
["BELONG", A, Q], ["BELONG", B, Q],
["BELONG", C, Q]
}

G' = {Q}

- ***Problems about determining attributes of Object in network of C-Objects***

Example 2: Problem about detemining attributes of right triangle in plane geometry. (Problem about solving right triangle)

Problem P2: "Give a right triangle ABC has A = 90^o, and AH is an altitude of triangle ABC.
Let AB = c, BC = a. Compute the length of AH."

\+ Specification of Sample Problem P2:

P2 = **L' → G'**

In which: L' = O' ⋃ F'

O' = { RightTriangle[A,B,C], Segment[A,H]}

F' = {["ALTITUDE", Segment[A,H],
RightTriangle[A,B,C]] ,
Segment[A,B].length = c ,
Segment[B,C].length = a
}

G' = {Segment[A,H].length}.

C. KBCO with Sample Problems model:

Definition 3.3: Knowledge Base of Computation Objects with Sample Problems (KBCO-SP) model consists of 7 components as followed:

$\mathcal{K}$ = (C, H, R, Ops, Funcs, Rules, Sample)

where:

- **(C, H, R, Ops, Funcs, Rules)** is knowledge domain which presented by KBCO model.
- **Sample** is a set of Sample Problems of this knowledge domain.

Kinds of facts and model of problem on KBCO-SP are defined similarly as above section and definition 2.2

KBCO-SP model is a model for designing practical knowledge bases, and designing algorithms to solve automatically problems based on a knowledge base. It is better in simulating the way of human thinking about searching solution of problem.

IV. THEOREM AND ALGORITHMS

A. Theorem:

Definition 4.1: Let (O, F) be a network of Com-objects, and M be a set of concerned attributes. Suppose A is a subset of M.

(a) For each f $\in$ F, denote f(A) is the union of the set A and the set consists of all attributes in M deduced

ISBN: 978-988-18210-9-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

from A by f. Similarly, for each Com-object $O_i \in O$, $O_i(A)$ is the union of the set A and the set consists of all attributes (in M) that the object O_i can determine from attributes in A.

(b) Suppose $D = [f_1, f_2, ..., f_m]$ is a list of elements in $F \cup O$. Denote:

$$A_0 = A,\ A_1 = f_1(A_0),\ A_2 = f_2(A_1)\ , \ldots.\ A_m = f_m(A_{m-1}) \text{ and } D(A) = A_m$$

We have $A_0 \subseteq A_1 \subseteq A_2 \subseteq \ldots \subseteq A_m = D(A) \subseteq F$

A problem $L \rightarrow G$ is called *solvable* if and only if there is a list $D \subseteq F \cup O$ such that $D(L) \supseteq G$. In this case, we say that D is a *solution* of the problem.

Definition 4.2: Given a model KBCO-SP $\mathcal{K}$ = (C, H, R, Ops, Funcs, Rules, Samples), and a problem $L \rightarrow G$ on this model. It is easy to verify that there exists an unique maximum set $\overline{L}$ such that the problem $L \rightarrow \overline{L}$ is *solvable*;

The set $\overline{L}$ is called the closure of L.

Lemma: Given a KBCO-SP model $\mathcal{K}$ = (C, H, R, Ops, Funcs, Rules, Samples), by definition 7, problem $L \rightarrow \overline{L}$ is solvable. Then, there exists an relation $D = [f_1, f_2, ..., f_k]$ such that $D(L) = \overline{L}$.

Theorem: Given a KBCO-SP model $\mathcal{K}$ = (C, H, R, Ops, Funcs, Rules, Samples). The following statements are equivalent.

(i) Problem $L \rightarrow G$ is solvable.

(ii) $G \subseteq \overline{L}$

(iii) There exists a list of relations D such that $D(L) \supseteq G$

Proof:

* The equivalent of (i) and (iii) can been checked easily.

* (i) => (ii): Problem $L \rightarrow G$ is solvable, but by definition 7, $\overline{L}$ is a maximum set such that problem $L \rightarrow \overline{L}$ is solvable, so $G \subseteq \overline{L}$.

* (ii) => (iii): Problem $L \rightarrow \overline{L}$ is solvable, by lemma, there exists an relation $D = [f_1, f_2, ..., f_k]$ such that $D(L) = \overline{L}$. But $G \subseteq \overline{L}$, so $D(L) \supseteq G$.

B. Algorithms for finding solution in KBCO-SP model:

Give the problem $P = L \rightarrow G$ as definition 2.2 on KBCO-SP, base on above theorem, the solution of problem P has been found though these steps:

Step 1: Classify problems such as problems as frames, problems of a determination or a proof of a fact, problems of finding objects or facts, etc.

Step 2: Classify facts and representing them based on the kinds of facts of KBCO-SP model.

Step 3: Modeling kinds of problems from classifying in step 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 following general algorithm represents one strategy for solving problems: forward chaining reasoning with heuristics and Sample Problems in which objects attend the reasoning process as active agents.

Step 1: Record the elements in hypothesis and goal part.

Step 2: *Find the **Sample Problem** can be applied.*

Step 3: Check goal *G*. If *G* is obtained then goto step 8.

Step 4: Using heuristic rules to select a rule for producing new facts or new objects.

Step 5: If selection in step 4 fails then search for any rule which can be used to deduce new facts or new objects.

Step 6: If there is a rule found in step 4 or in step 5 then record the information about the rule, new facts in Solution, and new situation (previous objects and facts together with new facts and new objects), and goto step 2.

Step 7: Else {search for a rule fails} Conclusion: Solution not found, and stop.

Step 8: Reduce the solution found by excluding redundant rules and information in the solution.

Some heuristic rules used in Step 4 have been mentioned in [22], so this paper only refers to search Sample Problem in Step 2. And it has been presented in next section.

C. Algorithms for finding Sample Problems:

Give the problem $P = L \rightarrow G$ on model KBCO-SP, the Sample Problem can be applied on P has been found by algorithm:

```
Step 1: SP ← Sample
        Sample_found← false
Step 2: Repeat
      Select S in SP
      if (facts of  L can be applied in problem S) then
        begin
          if kind of S.G’ = kind of G then
               Sample_found  ← true
          Else  if S.G’ ⊆ L then
               Sample_found  ← true
        end
      SP ← (SP – S)
   Until SP = {} or Sample_found

Step 3: if Sample_found then
          S is a sample problem of the problem;
        else
          There is no sample problem found;
```

This algorithm simulates a part of human mind when to find SP that relate to practical problem. Thereby, the inference of system has been more quickly and effectively. Moreover, the solution of problem is natural and precise.

D. Example:

Problem P: ”Let circle (O; 3) with center O and diameter BC = 6. Let A on this circle. H is projection of A on BC. Let AB = 4. Compute AH.”

\+ Active of Algorithms:

Step 1:

Firstly, we find the Sample Problem (SP) in **Sample** set can be applied to Problem P.

From two facts:

“ Segment BC is a diameter of circle (O,3)”

“ A is on circle (O,3)”

We use SP about “Determine Right Triangle” to determine ΔABC has $A = 90^o$

Step 2:

Check rules in **Rules** set, by using heuristics rules, we determine “AH is an altitude of ΔABC”

Step 3:

From the facts, "ΔABC is a right triangle", "AH is an altitude" and "BC = 6, AB = 4", we use SP about "Solve Right Triangle" in **Sample** set to determine the length of AH.

+ Comment:

In the inference processing, especially in Step 1 and Step 3, the system use SPs to deduce for problem. If we did not use Sample Problem, the system would check all rules in knowledge base, so the system found solution slowly. Using SPs improve the inference of system more optimal, make the system is more intelligent and immate the way of human's thinking better.

V. APPLICATION

Knowledge Base of Computational Objects model with Sample Problems has been applicated to represent some knowledge domains such as: Plane Geometry, Analytical Geometry, Linear Algebra, Physics (one-way electric and curcuit electric). In this section, based on exercise system in [13] and KBCO-SP model, we have built the system that was called *"The automatic solving plane geometry problems system"* by MAPLE and C#. This system can solve plane geometry problems in middle school automatically and the solution was more natural and closer with the way of thinking and solving of human.

A. Designing knowledge base of system:

KBCO-SP model conisists of seven components:

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

a) C–set of concepts of computational objects

The set *C* consists of concepts such as "Point", "Ray", "Segment", "Angle", "Line", "Triangle", "Trapezoid", "Circle", etc.

b) H–set of hierarchical relation on the concepts

From concepts of the objects introduced above, there are some hierarchical relations on them. Some of them are listed below:

- "Acute Angle", "Obtuse Angle" or "Straight Angle" is a special "Angle";
- "Isosceles Triangle", "Right Triangle " or "Equilateral Triangle" is also a "Triangle"

c) R–set of relations on Com-Objects

Between Com-Objects, there are various kinds of relations. We have some relations:

- Background relation: is relation of real numbers.
- Relations of basic level: are relations between basic objects and objects of first level.
- Relations of first level: are relations between basic objects, objects of first level and objects of second level, or relations between objects of higher levels.

d) Ops–set of relations on Com-Objects

In knowledge about plane geometry of middle shool, operators are relations between real numbers so we have:

Ops = {}

e) Funcs–set of functions on Com-Objects

The set Funcs consists of functions on C-Objects, such as:

+ Midpoint of a segment.
+ Projection of a point on a line,
+ Symmetrical point of a point through a line.

f) Rules–set of rules

Almost properties, clauses, theorems in plane geometry of middle shool can be represented by rules on facts relating to Com-Objects. Followings are some particular rules:

{a: segment, b: segment, c: segment,
a // b, c ⊥ a} ⇒ {c ⊥ b}

{A: point, B: point, C: point,
BC = AC} ⇒ {ABC is an isosceles triangle at C}.

{A: point, B: point, C: point,
AB ⊥ BC} ⇒ { angle ABC = 90°}.

g) Sample–set of Sample Problems:

+ Sample Problems about determining type of Objects, e.g: Right Triangle, Rectangle, Circle

+ Sample problems about:
  * Solving Right Triangle.
  * Solving Isosceles Triangle.
  * Relation between diameter and chord of circle.
  * Properties of Insrible Quadrilateral.

B. Designing inference engine of system:

Model of problem in this knowledge base about plane geometry is defined similarly as definition 2.2

(O, F, Goal)

Example:

Problem P: we use again problem P in section IV.D to illustrate the input and output of the system.

+ Model of this problem:

O = { Circle[O,3], Segment[B,C],
[A,Point], [H, Point] }

F = { ["BELONG",A, Circle[O,3]] ,
["DIAMETER", Segment[B,C], Circle[O,R]] ,
H = PROJECTION(A, Segment[B,C]) ,
Segment[B,C].length = 6;
Segment[A,B].length = 10;
}

Goal = {Segment[A,H].length}

Besides that, using algorithms in section IV.B and IV.C, the inference engine has been built. This engine simulates the way of human thinking about finding solution of practical problem by seraching sample problem can be applied in problem. This is like using the experience of human about practical problem to solve it.

+ The solution of this system:

Step 1:

{ ["DIAMETER", Segment[B,C], Circle[O,R]],
["BELONG", A, Circle[O,R]]
}

→ {RightTriangle[**A**,B,C]}

by "Deduce_SampleProblem" :
["Determine Right Triangle on cirlce"]

Step 2:

{H = PROJECTION(A, Segment[B,C]) }
→ {["ALTITUDE", Segment[A,H],
RightTriangle[A,B,C]] }

by "Deduce_Rules" :
["Properties of projection"]

ISBN: 978-988-18210-9-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Step 3:

{ ["ALTITUDE", Segment[A,H], RightTriangle[A,B,C]],
RightTriangle[**A**,B,C],
Segment[B,C].length = 6,
Segment[A,B].length = 4
}

→ {Segment[A, H] = $\frac{4}{3}\sqrt{5}$ }

by "Deduce_SampleProblem" : ["Solve Right Triangle"]

C. Compare this system with [24]:

KBCO-SP model is a significant improvement with model in [24]. It reflects more realistic about knowledge of human. This model has been used Sample Problems as experience of human for solving practical problems. Thereby helping the inference of system is faster and simulate the way of human thinking better.

Moreover, KBCO-SP model is also applied to design the knowledge base of plane geometry. This result is more advantages than the results of program in [24]. The solution of program is natural, and has logical reasoning similar to human reasoning.

VI. CONCLUSION

KBCO model is very useful and suitable for representing knowledge base, especially knowledge domains about Mathematics, Physics, Chemistry. Moreover, KBCO model has been extended by adding a Sample Problem component to the knowledge base and improving deduction techniques on this model, and the extension model is called *KBCO with Sample Problem* (KBCO-SP). The inference of the system becomes more intelligent and the solution of the system is more natural and similar to human's.

By using the KBCO-SP model, the program for solving plane geometry problems of middle school has been designed successfully. The solution given by this system is step-by-step natural, precise and has reasoning like human's.

In the future, we continue to research the KBCO-SP model to complete the program for solving problems on other knowledge domains, such as Analysis (one variable, multi variables), Physics (mechanics, optical) and Chemistry (organic and inorganic). In addition, the Computational Network and its application has been built successfully in [19, 21], moreover, model of Coputational Network using Sample Problems has been researched completely [17, 18]. Thus, the combination of the KBCO-SP model and Computational Network is expected to be an effectively model for knowledge representation.

REFERENCES

[1] Michel Chein & Marie-Laure Mugnier, 2009, *"Graph-based Knowledge representation: Computational foundations of Conceptual Graphs"*, Springer-Verlag London Limited.

[2] Frank van Harmelem & Vladimir & Bruce, 2008, *"Handbook of Knowledge Representation"*, Elsevier.

[3] Toshinori Munakata, 2008, *"Fundamentals of the New Artificial Intelligence: Neural, Evolutionary, Fuzzy and More"*, Springer-Verlag London Limited.

[4] F. Lehmann, 2008, *"Semantic Networks in Artificial Intelligence"*, Elsevier Science Ltd.

[5] M. Tim Jones, 2008, *"Artificial Intelligence: A System Approach"*, Infinity Science Press LLC.

[6] Asunción Gómez-Pérez, Mariano Férnandez-López, and Oscar Corcho, 2004, *"Ontological Engineering"*, Springer-Verlag.

[7] Stuart Russell & Peter Norvig, 2003, *"Artificial Intelligence – A modern approach (second edition)"*, Prentice Hall.

[8] Chitta Baral, 2003, *"Reasoning and Declarative Problem Solving"*, Cambridge University Press.

[9] John F. Sowa, 2000, *"Knowledge Representation: Logical, Philosophical and Computational Foundations"*, Brooks/Cole.

[10] George F. Luger & William A Stubblefield, 1998, "Artificial Intelligence", Addison Wesley Longman, Inc.

[11] G. Polya, 1997, *"How to solve it"*, VietNam Publisher of Education.

[12] Lakemeyer, G. & Nebel, B., 1994, *"Foundations of Knowledge representation and Reasoning"*. Berlin Heidelberg: Springer-Verlag.

[13] Vietnam Ministry of Education and Training, 2010, *"Textbook and workbook of plane geometry in middle school"*, Publisher of Education.

[14] Nhon Do, "Model for Knowledge Bases of Computational Objects", *IJCSI International Journal of Computer Science Issues*, Vol. 7, Issue 3, No. 8, pp. 11 – 20, May 2010.

[15] L. Stojanovic, J. Schneider, A. Maedche, S. Libischer, R. Suder, T. Lumpp, A. Abecker, G. Breiter, J. Dinger, 2010, "The Role of Ontologies in Autonomic Computing Systems", *IBM Systems Journal*, Vol . 43, No. 3, 2004

[16] Nhon Do, 2002, "Constructing intelligent systems for computation – Research and development of knowledge representation models to design systems for automated solving problems", *PhD. thesis*, University of Science, National University, Ho Chi Minh City, Vietnam, 2002.

[17] Nhon Do, Hien Nguyen, "A reasoning method on Computation Network and Its applications.", In *Proceeding of 2011 International MultiConference of Engineers and Computer Scientists,* (IMECS 2011), ISBN: 978-988-18210-3-4, pp. 137-141, Hongkong , March 2011.

[18] Nhon Do & Hien Nguyen, 2010, "Model for Knowledge Representation using Sample Problems and Designing a Program for automatically solving algebraic problems", In *Proceeding of World Academy of Science, Engineering and Technology*, (ICEEEL 2010), Issue 66, ISSN: 2070-3740, pp. 1436 – 1441, Paris, France, June 2010.

[19] Nhon Van Do & Tam Pham Huu, "Extensive Computational Networks And Applying in an Educational Software", In *Proceeding of 2009 International Conference on Artificial Intelligence and Education (ICAIE 2009)*, vol. 2, ISBN 978-1-84626-010-0, pp. 720 – 727, Wuhan, China, 2009.

[20] Nie Guihua, Jiang Xiangjie, Chen Donglin, Liang Yueling, Li Xiaofei, "The Research of Personalized Learning Based On Ontology", In *Proceeding of 2009 International Conference on Artificial Intelligence and Education (ICAIE 2009)*, vol. 1, ISBN 978-1-84626-010-0, pp. 22-26, Wuhan, China, 2009.

[21] Nhon Van Do, "Computational Networks for Knowledge Representation", *In Proceeding of World Academy of Science, Engineering and Technology (ICCSISE 2009)*, Volume 56, ISSN 2070 – 3724, Singapore, August 2009.

[22] Nhon Do, 2008, "An ontology for knowledge representation and Applications", In *Proceeding of World Academy of science, engineer and technology*, August 2008, vol. 32, ISSN: 2070-3740, pp. 23 – 31, Singapore, 2008.

[23] John F. Sowa, Arun K. Majumdar, "Analogical Reasoning", In *Proceeding of the International Conference on Conceptual Structures 2003*, pp. 16-36, Springer-Verlag , Germany, 2003

[24] Do Van Nhon, "A program for studying and Solving problem s in Plane Geometry", In *Proceeding of Conference on Artificial Intelligent 2000*, pp. 1441-1447, Las Vegas, USA, 2000.