World Academy of Science, Engineering and Technology
International Journal of Computer and Information Engineering
Vol:5, No:11, 2011

A Program for Solving problems in Inorganic Chemistry based on Knowledge Base

Nhon Van Do, Nam Hoai Le, Vien Chan Luong

Abstract—The Model for Knowledge Base of Computational Objects (KBCO model) has been successfully applied to represent the knowledge of human like Plane Geometry, Physical, Calculus. However, the original model cannot easyly apply in inorganic chemistry field because of the knowledge specific problems. So, the aim of this article is to introduce how we extend the Computional Object (Com-Object) in KBCO model, kinds of fact, problems model, and inference algorithms to develop a program for solving problems in inorganic chemistry. Our purpose is to develop the application that can help students in their study inorganic chemistry at schools. This application was built successful by using Maple, C# and WPF technology. It can solve automatically problems and give human readable solution agree with those writting by students and teachers.

Keywords—artificial intelligence, automated problem solving, knowledge base system, knowledge representation, reasoning strategy, education software/educational applications.

I. Introduction

A SUPPORTING program in education is really important for students and teachers. After recently surveying, many students agree that they need a computer program can help them in learning inorganic chemistry. The program have ability for reading problems and solving them automaticaly, then give students the results in the way they usually use in schools

To develop the program can solve problems automatically, the techniques of artifical intelligence, knowledge presentation and reasoning methods are very important and nessessary. In this case, the main programing tool is Maple (version 15), which is a symbolic programing language, and can help us in solving equations, logic expressions automatically. Morever, to make a program can read problems, show solutions like human and have friendly user interface, we have used C# language and WPF technology .

In this article, we present how to build a program has introduced above. First, we base on KBCO model [4] and make new extensions to build knowledge base used for inorganic chemistry field. Then, we write about the problem model and use design the reasioning methods to develop the inference engine. At last, we introduce something about the application and the example.

Our research have extended the KBCO model to apply in inorganic chemistry field, proposed new reasoning methods used in inference engine. The important extensions are type of attribute in Com-Object, Set computational relations and some new kinds of fact. We have developed the education application which can become a part of system that contain many supported applications in education. Morever, we hope our research will help us easy to apply KBCO model in others fields.

Nhon Van 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 The University of Natural 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.

II. Main Functions of Program

Main function of program is to solve some problems automatically in inorganic chemistry field. User only declare hypothesis and goal of problems by using a simple languague but strong enough for specifying them. The hypothesis need consist of objects (all objects appear in problems such as: elements, solutes, mixtures, gas or solid products in reactants etc...), and relations between them or between their attributes. It can alse contain formulas, relation formulas and some facts about the objects and their attributes. The goal need to be objects or attributes of objects which user want to identify or compute.

After specifying a problem, user click to a button to request the program to solve it automatically. The program will give a human readable result and solution when it finish solving. Of course, user can ask the program to save the problems and solutions for retrieval later.

III. Knowledge Representation

There are many methods and techniques ([7], [8], [1]) for knowledge representation, especially the KBCO model which was introduced in [4] and was successfully used in [2], [3], [5], [6]; is the most suitable to represent the Chemical knowledge domain and design inference engine for the solving problems, main feature described in section II.

A. Extensive Computational Object

The structure of a Com-object (modeled in [4]) is defined as a tuple 4 **(*Attrs*, *F*, *Facts*, *Rules*)** where $Attrs$ is a set of attributes, *F* is a set of equations called computation relation, *Facts* is a set of properties or event of object, and *Rules* is a set of deductive rule on facts.

1) Collection kinds attribute: Each attribute in $Attrs$ of a Com-object is one of two kind below:

Computational information
: like numeric (real, integer, complex, ...), boolean ...

Another lower level object
: for example Com-object `AB` represents a segment of

line has attribute `start` and `end` correspondingly to 2 objects `A` and `B` which represent 2 end points.

The second kind of attributes represents the *constructional relation* of two Com-objects. Because the $Attrs$ set is declared explicitly for a class of Com-objects due to the number of this kind relations also statically defined. But in the real world and particularly in Chemical field, there are exist objects that have the number of kinds of relations indefinitly. For example: a solution object mays have many substances as its solutes as possible.

Because of the above problem, beside the 2 kinds (1) *Computational information* and (2) *Lower level object*, a attribute of $Attrs$ may also is (3) a *Collection*. There is 3 kinds of a "collection" attribute can be:

- List — represent a ordered set of elements. `List<T>` tell that the attribute is a list of elements type `T`.
- Set — represent a unordered set of elements. `Set<T>` tell that the attribute is a set of elements type `T`.
- Dict — represent a key-value pair of dictionary. `Dict<TK, TV>` tell that the attribute is a dictionary where the keys is type `TK` and the values is type `TV`.

For example, a `MIXTURE` object has the attribute `Component` is type of `Set<SUBTANCE>` which represents the set of substances mixed in the mixture. A `EXPERIEMNT` object has the attribute `State` is `List<Set<SUBSTANCE>>` to recorded the states (each state is a set of substances) from the begining to the end of the experiment. Or the ρ attribute of a `SOLUTION` object is a `Dict<SUBSTANCE, real>` is the mole concentration table maps the substance s to it's mole concentration ρ_s.

2) Set computational relation: F is a set of computational relations. Each computational relation is simply a computational equation among the computational attributes inside a Com-object.

Corresponding the extension of attribute's kind, we introduce a new kind of computational relations call *Set computational relation*. There are two kind of *Set computational relation*:

Definition 1 (Set computational relation kind 1):

$$\forall x \in f(la_1, \ldots, la_n) \quad fr(x, na_1, \ldots, na_m)$$

where

- la_i — is a "collection" type attribute (for $1 \leq i \leq n$).
- f — is a function that return a (finite) set. f takes part of a filter, a combinator, ...
- na_j — is a computational value type attribute (for $a \leq j \leq m$).
- fr — is a fuction that return a formula relations.

An example of set computational relation kind 1:

$$\forall \{x, y\} \in \binom{S}{2}, \quad \frac{r_x}{r_y} = \frac{n_x}{n_y}$$

Notation: with S is a set and k is a positive interger then $\binom{S}{k} = \{X : X \subseteq S \text{ and } |X| = k\}$

Definition 2 (Set computational relation kind 2):

$$fr(la_1, \ldots, la_n, na_1, \ldots, na_m)$$

where

- la_i — is a "collection" type attribute (for $1 \leq i \leq n$).
- na_j — is a computational value type attribute (for $a \leq j \leq m$).
- fr — is a function that return a formula relation. fr usually contain the set operator like $\sum$, $\prod$, $\bigcup$...

An example of set computational relation kind 2:

$$m_{\text{mixture}} = \sum_{s \in Component} m_s$$

3) Com-Object Structure: The structure of a Com-object can be modeled by $(Attrs, F, Facts, Rules)$ where

- $Attrs$ is a set of attributes. $Attrs$ can be partitioned to three seperate sub-collections

  $$Attrs = Attrs_N \bigcup Attrs_O \bigcup Attrs_S$$

  where
  - $Attrs_N$ — is a set of computational kind attritbutes.
  - $Attrs_O$ — is a set of lower level object kind attributes.
  - $Attrs_S$ — is a set of collection kind attributes.
- $F = F_E \bigcup F_S$ is a set of computational relations which is constructed from two subset
  - F_E — is a set of equation kind of computational relations.
  - F_S — is a set of set computational relations. It is combined from 2 seperate subset F_{S_1} and F_{S_2} corresponding two kind of set computational relation.
- *Facts* is set of properties or event of objects
- *Rules* is set of deductive rules on facts.

For example, a model for class of a `SOLUTION` Com-object are as follows:

$Attrs = Attrs_N \bigcup Attrs_O \bigcup Attrs_S$

$Attrs_N = \{m_{Solution}, V, m_{Solutes}, pH, \rho_0\}$ (solution's mass, volume of solution, solutes's mass, mass density).

$Attrs_S = \{Solutes, c, \rho, w\}$ where $Solutes$ is type `set(SUBSTANCE)`, is a set of dissolved substance; c (ρ, w) is molar concentration (mass concentration, mass fraction), is type of `dict(SUBSTANCE` $\rightarrow$ `NUMERIC)`. For each dissolved `SUBATANCE` object s in this `SOLUTION` object then c_s is molar concentration of s in solution.

$Attrs_O = \{Solvent\}$ where $Solvent$ is a `SUBSTANCE` object.

$F = F_E \bigcup F_S$

$F_E = \left\{\rho_0 = \frac{m_{Solution}}{V}\right\}$

$F_S = F_{S_1} \bigcup F_{S_2}$

$$F_{S_1} = \left\{ \begin{array}{c} \forall s \in Solutes\, c_s = \frac{n_s}{V}, \\ \forall s \in Solutes\, \rho_s = \frac{m_s}{V}, \\ \forall s \in Solutes\, w_s = \frac{m_s}{m_Solution}, \\ \forall s \in Solutes\, c_s = \frac{\rho_s}{M_s}, \\ \forall s \in Solutes\, w_s = \frac{\rho_s}{\rho_0} \end{array} \right\}$$

$$F_{S_2} = \left\{ \begin{array}{c} \rho_0 = \sum_{s \in Solutes} \rho_s, \\ m_{Solutes} = \sum_{s \in Solutes} m_s \end{array} \right\}$$

$Facts = \{m_{Solution} > 0, \rho_0 > 0, V > 0, m_{Solutes} > 0\}$
$Rule = \{\ldots\}$

4) Behaviors: As a major characteristics, a Com-object must be equipped abilities to solve problems by applying the internal dedutive information *F*, *Rules* with the current state of the attributes. Because of the structure's extension of Com-object, the behaviors of object must also be able to handle the new *Set computational relations*. The algorithms 1 and 2 is the simple and naive appoarch algorithm to deduce information from a *Set computational relation* f.

Algorithm 1 Deduce from $f \in F_{S_1}$

1: **if** la_i is determined $\forall i$ **then**
2: $C \leftarrow f(la_1, \ldots, la_n)$
3: **return** $\{fr(x, na_1, \ldots, na_m) \quad \forall x \in C\}$
4: **end if**

Algorithm 2 Deduce from $f \in F_{S_2}$

1: **if** la_i is determined $\forall i$ **then**
2: **return** $fr(la_1, \ldots, la_n, na_1, \ldots, na_m)$
3: **end if**

For example, when a `SOLUTION` object has $Solutes = \left\{\mathrm{Na}^+, \mathrm{Ba}^{2+}, \mathrm{OH}^-\right\}$ then when apply the $f = \left(\forall s \in Solutes\, c_s = \frac{n_s}{V}\right)$ will give us 3 equation kind formular relations $\left\{c_{\mathrm{Na}^+} = \frac{n_{\mathrm{Na}^+}}{V}, c_{\mathrm{Ba}^{2+}} = \frac{n_{\mathrm{Ba}^{2+}}}{V}, c_{\mathrm{OH}^-} = \frac{n_{\mathrm{OH}^-}}{V}\right\}$.

B. Chemistry's Knowledge Model

1) Components of model: The Chemistry's Knowledge model is based on the KBCO model with the Extensive Computational object model in III-A.

Definition 3 (Chemistry's KB model):

$$(C, H, R, Rules)$$

- C is a set of concepts of Com-object. Each concepts is a class of Com-objects.
- H is a set of hierarchy relation on the concepts.
- R is a set of relations on the concepts.
- $Rules$ is a set of rules.

The concepts in C include the `SUBSTANCE` class and its derived, the `ELEMENT`, `SOLUTION`, `MIXTURE`, `REACTION` and `EXPERIMENT` class. The H represent the ordered relation "specialization" of the concepts (especially the `SUBTANCE` and its subclass) in C.

The R other kind relation on C. For example the "similar" relation of 2 solutions objects tell the objects has the same kind of solutes, same of mole concentration for each solute. And finally the R represent for deductive rules usually in form **`if`** `hypothetical-facts` **`then`** `conclusive-fact`

2) Kinds of Facts: Because the lacking of components to the general KBCO model in [4], the requirement of program does not meet all of 11 kinds in [4] except the first 6 kinds of facts do be useful. Beside that, there are more 4 new kinds of facts that necessary to define for the chemistry's domain. The kinds of facts are as follows: (the first 6 is derive from [4]).

- **Fact of kind 1**: information about object kind, like H_2SO_4 is a acid.
- **Fact of kind 2**: a determined of an object or an attribute of an object.
- **Fact of kind $\bar{2}$**: a non-determined of an object or an attribute of an object.
- **Fact of kind 3**: a determined of an object or an attribute of an object by a value or a constant, for example: $n_{\mathrm{Ba(OH)_2}} = 0.5$.
- **Fact of kind 4**: equality on object or attributes of object, for example: $n_{\mathrm{FeO}} = n_{\mathrm{Fe_2O_3}}$.
- **Fact of kind 5**: a dependence of an object on other object by a general equation. For example: $n_{\mathrm{OH}^-} = n_{\mathrm{NaOH}} + 2n_{\mathrm{Ba(OH)_2}}$.
- **Fact of kind 6**: a relation on objects or attributes of the objects. For example: solution X is "similar" with solution Y.

Following is the 4 new kinds of facts.

- **Fact of kind 7**: a comparison of an attribute with a constant, like $n_{\mathrm{H}^+} > 0$.
- **Fact of kind 8**: a determined of the domain of an object or an attribute. For example: element $M \in \{\mathrm{Na}, \mathrm{Ba}, \mathrm{Ca}, \mathrm{K}\}$.
- **Fact of kind 9**: a fact of a collection kind attribute is a subset of a constant set or another collection kind attribute. For example: reaction r_1's attirbute $Reactants$ is **subset** of experiment e's current state $State_0$.
- **Fact of kind 10**: a determined of the component of a collection kind attribute. For example: the solution X's $Solutes$ attribute containts 2 kinds of salt. This fact has 3 parameter a, n, c where a is the collection kind attribute (X's $Solutes$); c is a concept in C; n can be a positive integer (in the example, $n = 2$), 0 (mean there are no c object in a) or `none` (mean there are c objects in a but the number of c objects is unknown).

IV. Modeling Problems & Reasoning

A. Problems Model

The *Network of Computational Objects* in [4] is use as a model of the problems; and be able to specificate almost problems in Chemistry high school education program. The structure of problem model is denoted by:

$$(O, F) \rightarrow G$$

- O is a set of Com-objects.
- F is a set of facts or computational relations among the objects in O.
- G is a goal set, it consist the target attributes, objects that need to find.

For example:

Completely absorbed 4.48 liters of CO_2 (at TOR) in 500 ml solution of NaOH 0.1 M and $Ba(OH)_2$ 0.2 M, generated m gram precipitate. The value of m is:
A. 19,70. B. 17,73. C. 9,85. D. 11,82.

The above problem can be considered on the network of Com-objects (O, F) as follows:

$$O = \left\{ \begin{array}{rl} subCO2 & \texttt{SUBSTANCE } CO_2 \\ subNaOH & \texttt{SUBSTANCE NaOH} \\ subBaOH2 & \texttt{SUBSTANCE } Ba(OH)_2 \\ solX & \texttt{SOLUTION} \\ e & \texttt{EXPERIMENT} \\ objP & \texttt{unknown} \end{array} \right\}$$

$$F = \left\{ \begin{array}{c} subCO2_V = 4.48 \\ solX_{Solutes} = \{subNaOH, subBaOH2\} \\ solX_V = 0.5 \\ solX_{MC}[subNaOH] = 0.1 \\ x_{MC}[subBaOH2] = 0.2 \\ e_{Input} = \{subCO2, solX\} \\ objP = e_{SolidOutput} \end{array} \right\}$$

and goal $G = \{objP_m\}$.

B. Inference modules

There are 4 important modules in the inference engine: `Apply Rules`, `Deduce Inside Objects`, `Deduce computational relations` and a bundle of deduction methods.

The `Apply Rules` is try to apply the rules in R. The module try to match the current facts with the hypothetical-facts of the rule. If all the hypothetical-facts are satisfied, then we get the conclusive-facts and record the case. If there exist at least one hypothetical-fact that inconsistent with the current facts, then in this case, the rule is never meet, we also record the case. Recording the case help to avoid multiple-time applying the same case to the rule. If the module can not determined the case is satisfied or not, it ignore the case in this time.

The `Deduce inside objects` module is try to use the solving problems behavior of Com-object to get more deduction. Also the `Deduce computational relations` is try to apply the computational relations among the objects.

V. Application

The program for solving problems in inorganic chemistry was designed include:

- Four main components of program is *knowledge base, inference engine, user inteface and utility functions*. Knowledge base contain all of definations of inorganic chemistry. Each defination is a Com-Object such as: element, substance, mixture, solution etc... Inference engine or knowledge processing module is written in Maple, it has to read knowledge base, read problems and find the solution automatically. User interface is a window for inputting problems above special language and another window for showing solutions of problems and figures. Utilify functions support in processing special data in chemistry as the chemistry formula, the solution table or the periodic.
- The program require users declare hypothesis above special language, it's very simple and strong enough for specifying problems. The hypothesis can consist of objects, relations between objects or attributes. The goal can be to compute the attributes. After specifying problem, user clicks a button to request program solving problems automatically. After finishing, the program will give users the solutions and results. They are easy to read and understand by many students and teachers becouse of using a simple language and show solutions steps by steps.

Example: continue the previous example problem, the Solution found by the system as follow:

`SUBSTANCE` objects information:

Object	Formular
subCO2	CO_2
subNaOH	NaOH
subBaCO3	$BaCO_3$
subBaOH2	$Ba(OH)_2$
subBa2b	Ba^{2+}
subBa2a	Ba^{2+}
subCO32	CO_3^{2-}
subHCO3a	HCO_3^-
subHCO3b	HCO_3^-
subNa	Na^+
subOHa	OH^-
subOHb	OH^-

`SOLUTION` objects information:

Object	Solutes
solX	subNaOH, subBaOH2
	subBa2a, subNa, subOHa
solOut	subBa2b, subHCO3b, subNaS

`EXPERIMENT` object e

Input = { subCO2, solX }

State 1 = { subCO2, subBa2a, subNa, subOHa }

CO_2	0.2(mol)	from (2)
Ba^{2+}	0.1(mol)	from hypothesis
Na^+	0.05(mol)	from hypothesis
OH^-	0.25(mol)	from (7)

Phase 1 = { r1 }

$$\begin{array}{ccccc} CO_2 & + & OH^- & \rightarrow & HCO_3^- \\ x_1 & & x_1 & & x_1 = 0.2 \end{array}$$

State 2 = { subBa2a, subHCO3a, subNa, subOHb }

Ba^{2+}	0.1(mol)	from hypothesis
HCO_3^{1-}	0.2(mol)	from hypothesis
Na^+	0.05(mol)	from hypothesis
OH^-	0.05(mol)	from (6),(9)

Phase 2 = { r2 }

$$\underset{x_2}{HCO_3^-} + \underset{x_2}{OH^-} \rightarrow \underset{x_2}{H_2O} + \underset{x_2 = 0.05}{CO_3^{2-}}$$

State 3 = { subBa2a, subCO32, subHCO3b, subNa }

Ba^{2+}	0.1(mol)	from hypothesis
CO_3^{2-}	0.05(mol)	from (11),(13)
HCO_3^{1-}	0.15(mol)	from hypothesis
Na^+	0.05(mol)	from hypothesis

Phase 3 = { r3 }

$$\underset{x_3}{Ba^{2+}} + \underset{x_3}{CO_3^{2-}} \rightarrow \underset{x_3 = 0.05}{BaCO_3}$$

State 4 = { subBaCO3, subBa2b, subHCO3b, subNa }

$BaCO_3$	0.05(mol)	from (15), (16)
Ba^{2+}	0.05(mol)	from hypothesis
HCO_3^{1-}	0.15(mol)	from hypothesis
Na^+	0.05(mol)	from hypothesis

Output Solid = subBaCO3
Output Solution = solOut

The solving steps of the problems:

(1) reaction r1 is complete $\rightarrow n_{subsCO2} = x_1$

(2) $V_{subCO2} = 4.48$ and $V_{subCO2} = 22.4 * n_{subsCO2} \rightarrow n_{subsCO2} = 0.2$

(3) $V_{solX} = 0.5$ and $\rho_{solX}^{subBaOH2} = 0.2$ and $\rho_{solX}^{subBaOH2} = n_{subBaOH2}/V_{solX} \rightarrow n_{subBaOH2} = 0.1$

(4) $V_{solX} = 0.5$ and $\rho_{solX}^{subNaOH} = .1$ and $\rho_{solX}^{subNaOH} = n_{subNaOH}/V_{solX} \rightarrow n_{subNaOH} = 0.05$

(5) $n_{subsCO2} = 0.2$ and $n_{subsCO2} = x_1 \rightarrow x_1 = 0.2$

(6) after reaction r1 $\rightarrow n_{subOHb} = n_{subOHa} - x_1$

(7) $n_{subNaOH} = 0.05$ and $n_{subBaOH2} = 0.1$ and $n_{subOHa} = n_{subNaOH} + 2 * n_{subBaOH2} \rightarrow n_{subOHa} = .25$

(8) reaction r2 is complete $\rightarrow n_{subOHb} = x_2$

(9) $n_{subOHa} = .25$ and $n_{subOHb} = n_{subOHa} - x_1$ and $x_1 = .2 \rightarrow n_{subOHb} = 0.05$

(10) $n_{subOHb} = 0.05$ and $n_{subOHb} = x_2 \rightarrow x_2 = 0.05$

(11) after reaction r2 $\rightarrow n_{subCO32} = x_2$

(12) reaction r3 is complete $\rightarrow n_{subCO32} = x_3$

(13) $n_{subCO32} = x_2$ and $x_2 = 0.05 \rightarrow n_{subCO32} = 0.05$

(14) $n_{subCO32} = 0.05$ and $n_{subCO32} = x_3 \rightarrow x_3 = 0.05$

(15) after reaction r3 $\rightarrow n_{subBaCO3} = x_3$

(16) $n_{subBaCO3} = x_3$ and $x_3 = 0.05 \rightarrow n_{subBaCO3} = 0.05$

(17) $M_{subBaCO3} = 197$ and $n_{subBaCO3} = 0.05$ and $n_{subBaCO3} = m_{subBaCO3}/M_{subBaCO3} \rightarrow m_{subBaCO3} = 9.85$

VI. Conclusion

The KBCO model with extensions about type of attribute in Com-Object, kinds of fact and kind of relation give us more confortable when we represent knowledge on the computer. Beside, problem model and reasoning althorithms were upgraded to use in new extensions easily. The extensions of model and althorithms are used to apply in solving problems automatically in inorganic chemistry field successfully. We have built a program which can read problems, solve problems automatically and give the result in the way like people. In testing period, the program has solved many common problems in inorganic chemistry rapidly.

The program was built in Maple, C# language and WPF technogory, has a friendly user interface and easy to use by students and teachers. Our purpose in furture is continue to upgrade the model and althorithms to solve more problems. This methods can be used for another programs for studying knowledge and solving problems in the others fields.

References

[1] Chitta Baral. *Knowledge Representation, Reasoning, and Declarative Problem Solving*. Cambridge University Press, Cambridge, England, 2003.

[2] Nhon Van Do. A Program for studying and Solving problems in Plane Geometry. In *International Conference on Artificial Intelligence 2000*, pages 1441–1447, 2000.

[3] Nhon Van Do. A system that supports studying knowledge and solving of analytic geometry problems. In *16th World Computer Congress 2000, Proceedings of Conference on Education Uses of Information and Communication Technologies*, pages 236–239, Beijing, 2000.

[4] Nhon Van Do. Model for Knowledge Bases of Computational Objects. *IJCSI International Journal of Computer Science Issues*, 7(3):11–20, 2010.

[5] Nhon Van Do, Hung Kim Chau, and Van Long Ho. An Intelligent Educational Software for Automatic Problem Solving in Linear Algebra. In *The 6th International Conference on Computer Science & Education*, number Iccse, pages 697–702, Singapore, 2011.

[6] Nhon Van Do and Thu Le Pham. Knowledge Representation and Algorithms Automatic Solving Integral Problems. In *The 6th International Conference on Computer Science & Education*, number Iccse, pages 730–735, Singapore, 2011.

[7] George F. Luger and William A. Stubblefield. *Artificial Intelligence: Structures and Strategies for Complex Problem Solving*. Benjamin/Cummings, Redwood City, California, second edition, 1993. Luger, Geoge F. and Stubblefield, William A.

[8] Stuart Russell and Peter Norvig. *Artificial Intelligence: A Modern Approach*. Prentice Hall, 1995.