과학실험에서의 모델 설계 및 구현

Design and Implementation of Science Experiment Models for Artificial Chemistry Laboratory

변 영 태
(Young-Tae Byun)

요 약 과학 분야의 교육에서 실험이 차지하는 비중은 다른 분야에 비하여 매우 크다. 과학실험은 여러 가지 실험구성요소 선택과 배치 및 적절한 실험 행위에 의하여 어떤 실험 상황이 만들어지고 자연법칙에 의해 상태변화가 발생하여 실험을 진행하게 된다. 과학자 혹은 학생이 실험을 하면서 그 실험에 대한 인지 모델을 내부적으로 구축하고 그 모델을 사용하여 시스템행위에 대한 인과추론을 함으로써 실험을 통해 진리 학습과 탐구를 하게 된다. 2차원 모델과 인과추론에 필요한 본체론적 엔티티의 표현과 더불어 실험에서의 변화를 유도하는 추론방법을 본 연구에서 제시한다. 구현을 위해 화학영역의 실험을 선택하였고, 실험구성요소 (실험 기구와 시약) 및 물리적, 화학적 자연법칙 뿐만 아니라, 실험시스템 구조와 각 실험요소의 기능을 고려한 추상지식까지를 지식표현 대상으로 하고 있는 것이 본 연구의 특징이다. 학생이 임의의 실험을 할 수 있고 실세계와 유사한 현상을 보이는 가상 화학실험실을 개발하여 제안된 방법의 적정성을 확인하였다.

Abstract We believe that science experiments in a laboratory are essential for science education. Scientific experiments begin with situations set by selecting and locating tools and reagents, and by proper experimental behavior, and thereafter situations are changed by natural laws and intermediate experimental behavior. While scientists and students do experiments, they build a cognitive model internally, do causal reasoning on the model to derive system behavior, and then learn scientific truth. We suggest not only a representation method for a 2-dimentional model and for ontological entities necessary in causal reasoning, but also an inferencing method to derive behavior. Chemistry experiments are chosen for the implementation. For the ontological entities, we consider experimental tools, reagents and their heirarchical structures, physics and chemistry natural laws, and functional abstraction knowledge. In order to show the usefulness of our methods, we have developed a program, called ACL(Artificial Chemistry Laboratory), which provides an experiment environment where students can do non-predetermined experiments, and shows experimental system behavior similar to what happens in the same situation in a real world and descriptions about why it happens.

1. Introduction and Related Work

Experiments in a laboratory are very important in science education for primary, middle, and high school students. It may be very interesting to identify what kinds of knowledge and cognitive models are used by scientists and students and to find out how they understand natural laws and facts and how they reason about scientific experiments. After understanding their mind, it may be helpful to develop a computer program which imitates the real world and a reasoning mechanism so that the program is used by students to learn scientific knowledge by doing artificial experiments. Once we identify what there exist in scientist's mind, we need to

---

The author wishes to acknowledge the financial support of the Korea Research Foundation made in the program years of 1996 - 1999.

represent them in a proper way, and then to build in a computer a knowledge base using a proper knowledge representation method and an inference engine so as to derive phenomena similar to what happen in the same situation in a real world. The program needs to maintain internally a correct model corresponding to a cognitive model.

Using an internal model and information of the knowledge base, the inference engine should be able to infer what phenomena or changes should happen in a real world and to update properly the model corresponding to those changes. If only predetermined experiments were allowed in a system, there was no need to maintain an internal cognitive model. It would be enough to keep the necessary information for each experiment case. When there are more than 200 reagents and we consider experiments using three reagents, we need to compile several millions cases in the system beforehand, and we know it is almost impossible. Therefore, maintaining an internal model corresponding to an experiment situation is necessary for non-predetermined experiments.

Traditional modeling and simulation methods [10] usually build models before simulation and change only parameters during simulation, and only exact or approximate quantitative equations are used in them. They are improper for maintaining variant models for dynamic environments. One research area about the modeling and simulation of a physical world is Qualitative Physics and Reasoning [1, 7, 9]. The research groups represent models and compute system behavior mainly in common sense. Some rules of natural phenomena are known quantitatively and others are not. Although complicated quantitative equations are known, it may cost much time for exactly correct computation and it may be unnecessary in many cases. For instance, a human being reasons about physical systems successfully without exactly correct quantitative equations. We may employ many natural laws with quantitative equations, but not in every case where a highly complicated mathematical formula does not have to be presented to the last degree when we consider simulation effectiveness. [8] showing a qualitative method to handle particle's movement and effect is a good example for this argument. As far as models and reasoning are concerned, there are many researches like [4, 13] which concentrate on diagnostics, and [2, 5] which propose methods for the composition of models and the derivation of system behavior. Domain problems of their work are different from ours.

As far as the program that we develop is concerned, we need to point out two important things. Firstly, experiments in a laboratory are very important in science education. However, education systems in many countries do not provide enough experiment chances to students for several reasons. The reasons are the insufficiency of financial support from a government, heavy load to teachers, danger of experimental accidents, heavy pressure to get high scores of entrance written exams for a school, and etc. Even if students had reasonable experimental chances, every student didn't participate equally in experiments and they did only predetermined experiments during class hours. Nobody can do any experiment, which is not predetermined, whenever they want. The proposed program can be used by a student as an alternative to solve the problem. Secondly, there are many science courseware systems, but most of them provides only "just-seeing" but not "just-doing". [15] discusses the problems of "just-seeing" and the importance of "just-doing". The program ACL can provide this facility. Several systems like [3, 12, 14] have a function like electronic textbooks, or characteristics similar to the program, but we believe they do not handle non-predetermined experiments successfully.

We discuss in Section 2 what kinds of

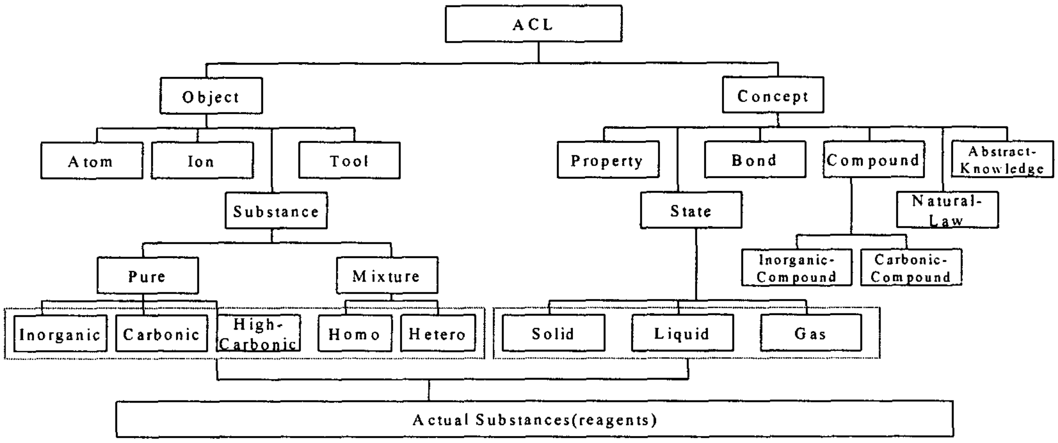


Figure 1. A Hierarchical Structure for the ACL Knowledge Base

ontological entities need to be identified and how they are represented, and internal models describing experimental situations along with spatial descriptions. Tangible things in Section 3, intangible things and inferencing in Section 4, implemenation and results in Section 5, and summary in Section 6 are discussed in this paper, respectively.

2. Ontological Entities and Model Descriptions

There can be two groups of ontological entities in scientific experimental reasoning. The first is a group of tangible things such as atoms, pure substances and tools, and the second is a group of intangible things such as existing states of reagents, substance properties, chemical knowledge and natural laws. Figure 1 shows a hierarchical sturucture of knowledge base consisting of the two groups.

The *Object* class in the figure is defined to generally represent objects which are physically tangible things existing in nature [11]. This class itself is a subroot leading another hierarchy, and we discuss this class in detail in Section 3. The *Concept* class in the knowledge base is for intangible things. In ACL, various chemistry and physics laws are represented by functions, called *processes*. The subhierarachical structure rooted by the *Concept* class is also classified into subclasses characterizing different conceptual information on chemistry, physics, and other abstract knowledge. We discuss this class in detail in Section 4.

We believe that an existing substance in a real world is made by composing a substance and an existing state. For instance, water is made by composing a reagent H2O and the liquid state. The Actual Substance class shown at the bottom of the knowledge base refers to any actually existing substances, called as reagents.

In other word, the reagents are instances of the *Actual Substance* class. The *Actual Substance* class multiply inherits attributes and methods of classes belonging to the object hierarchy and those to the concept hierarchy, respectively. In this way, any existing substances or reagents in ACL have two ways of access to object-related and conceptual information. [6] discusses several different existing forms of water, and some of them are available in our system.

There are two kinds of things involved in an experiment. They are tools and reagents, and we use a term, *experimental components*, for them in ACL. To simulate experiments and enable inferences, an internal model containing information about selected experimental components and their spatial relations must be maintained.

We use a term, LB(*Lab Board*), where the internal model representing an experiment situation is saved. The internal model can be regarded as a directed graph where nodes correspond to experimental components and edges have spatial information between nodes. The inference engine of ACL uses the internal model to infer what should happen in a situation. LB is further divided into ToolLabBoard for tools and ReagentLabBoard for reagents. The information of the spatial relation between experimental components is attached on directed edges connecting two experimental components. Vocabularies that describe spatial relation are in Table 1, and Figure 3 shows an example of the spatial relations when experimental components are arranged as in Figure 2.

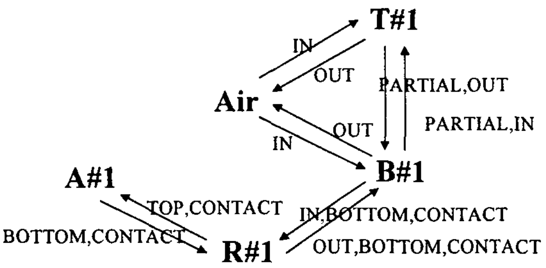


Figure 1. A Directed Graph for the Situation in Figure 2.

IN: inner space of an experimental component
OUT : outer space of an experimental component
TOP : upper space ″
ABOVE : above space ″
BOTTOM : lower space ″
BELOW : below space ″
IO1 : connected to the first I/O port of ″
IO2 : connected to the second I/O port of ″
HOMO-MIX : homogeneously mixed with others
HETERO-MIX : heterogeneously mixed with others
CONTACT : contact with other
PARTIAL : partially related with others and used with IN or OUT

Table 1. Vocabularies for Spatial Information

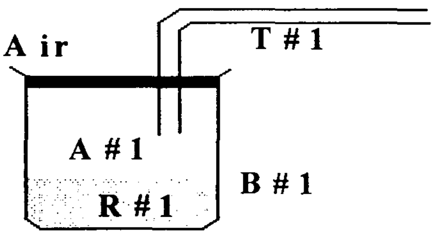


Figure 2. A Situation
(Air: Outer air, A#1: Inner air, R#1: Liquid reagent, B#1: Beaker, T#1: Glass tube)

3. Tangible Things

There are several classes in our work, which occupy space and have weight, and we discuss these in this section.

3.1 Atoms and Ions

In ACL, there are 103 atoms and their attribute values such as *Atomic_Symbol, Atomic_Number, Group, Period, Radius, Weight, Volume, Metality, Oxidation_Number, s,p,d,f Orbit_Information, and etc.* Although it is true that an atom is divided into smaller particles, we follow in ACL the theory of Dolton that the atom is inseparable.

When an atom or a molecule gets or loses an electron or electrons, it becomes charged and called an ion. There are single and multiple ions (a.k.a. radical ions). The attributes of the classes are *Ionic_Symbol, Oxidation_Number, Weight, Component_List, and pKa or pKb*. The atom information is also used for a single ion instance.

3.2 Tools

There are more than 20 experiment tools provided in ACL. From the functional point of view, we group these into 6 types. Each type has its own properties.

Container :	Beaker(small or big), Test Tube, Evaporating Dish, Erlenmeyer Flask, Florence Flask, Gas Tank
Dropper :	Separatory Funnel, Dropping Pipet
Connector :	Glass Tube, Hose, Drying Tower, Liebigh Condenser
Litmus :	Red Litmus paper, Blue Litmus paper
Filter :	Buchner Funnel
Heater :	Alcohol Lamp, Candle, Match

Table 2. Experimental Tools in ACL

3.3 Substance Class and Subclasses

The Substance class is a child of the Object class, and a parent of the Pure class and the Mixture class. The Pure class is further divided into the Inorganic class and the Carbonic class, and the Mixture class is divided into the HomoMixture class and the HeteroMixture class. The Substance has many attributes such as *Symbol, State, Substance_Type, Component_List, Component_Rate, Weight, Volume, Density, Compound_Type_List, Subst_ProcessId_List, Color, and etc.* Each instance has a list of processes which specially belong to this instance.

The Pure class is a subclass of the Substance class. It inherits all attributes of the Substance class, and has unique values for attributes like *Bondtype, Melting_Point, pKa, Boiling_Point, Solubility, Ionization_Degree, Polarity and etc.*

Inorganic reagents are instances of the Inorganic class which inherits the attributes of the Pure class and the Substance class. It has the Compound_type_list attribute for compound information. For the compound information, we have the Compound class in the right of the knowledge base. A carbonic reagent is an instance of the Carbonic class which is a subclass of the Pure class and the Substance class. It also inherits all attributes of super classes and has a value for the Compound_type_list attribute. The Highcarbonic class is for a substance as polystyrene consisting of molecules that are large multiples of units of low molecular weight.

3.4 Homogeneous and Heterogeneous Mixture

The Mixture class is for tangible things in the mixed state of two or more pure materials, and divided into homogeneous and heterogeneous by the mixed state. When the components of mixture are mixed uniformly, we call it homogeneous mixture, and otherwise heterogeneous mixture. It has attributes such as *Boiling_Point, Melting_Point, and Polarity.*

The attributes of pure reagents are defined uniquely, whereas those of mixture reagents cannot. For example, the weight of homogeneous mixture is the sum of that of components. The attributes of homogeneous mixture is closely related with that of components. Homogeneous mixture also has its own spatial relation. It and its components have the same spatial relation, because they are mixed uniformly. The spatial relation is systematically managed by a method so that homogeneous mixture and its components hold the same spatial relation.

In ACL, water solution is treated as the special case of liquid homogeneous mixture. The reason is that water solution is used frequently in real experiments and have many important properties. A reagent can be solved by water, and an ion can be created by solution. Becoming acid or base is also happened in the water solution.

Reagents such as a gold-plate ring or a loaf of bread are examples of solid heterogeneous mixture. When a solid heterogeneous mixture reagent is created, it has several components. The internal structure of solid heterogeneous mixture uses a list of a solvent and solutes.

One liquid reagent and one or more

components can make liquid heterogeneous mixture reagents. The liquid heterogeneous mixture does not have the same structure as the solid one. It is handled by only spatial descriptions.

4. Intangible Things and Inferencing

There are several ontological entities of intangible things, and we identify existing states, properties, types of chemical bonds, types of compounds, natural laws, and functional abstract knowledge about tools. Some of them are already discussed briefly in previous sections and we discuss in this section the most important intangible things which are the states, the natural laws and the abstract knowledge.

4.1 Existing States

Every pure existence in a real world has its own state: Solid, Liquid, or Gas. Since each state has its own special characteristics, we classify these states in ACL. Each state class includes relevant natural laws. For instance, a process "Solid stays on the bottom of an empty beaker," and a process "Solid sinks into liquid when the density of the solid is greater than that of the liquid" are attached to the solid class. Actual substance instances are created with multiple inheritance of one of substance classes and one of the state classes. The natural law class in the right half of the knowledge base has several physics laws for states.

However, this is not enough to describe a reagent existing in a real world, because there are two or more shapes of the same state. For instance, a solid reagent can exist as loaf or powder. Therefore we add the shape information to solid and liquid state reagent. For solid, *Loaf*, *Powder*, *Coating*, and *Amorphous* are used, and for liquid, *Dispersed* and *Normalshape* are used, respectively. Examples using the shape information are that the water in fog has a shape description *Dispersed*, that the outside solid reagent of oxidized solid has a shape description *Coating*, and that the dust in air has a shape description *Amorphous*. Since a gas exists only as molecules, there is no special shape description.

4.2 Processes

In ACL, we use a term, *processes*, as the entities that describe natural laws to derive experimental behavior and abstract knowledge about tools. Every process has a precondtion and condition part and an action part. Several process examples are shown below. Compared to normal production rules, there is one more part which is a precondition part. This part is not actually coded in a process. It just means that, only when an experimental component is selected and used in an experiment, corresponding processes related to the experimental component become candidtes to be checked by the inference engine. Processes related to non-participating experimental components are never checked at all by the engine. All processes used in ACL are classified into 1) processes related tools, 2) processes related to reagents according to facilities and attributes, and 3) processes for physics laws.

4.2.1 Processes Related to Tools

Since each tool has its own purpose and does its function in a certain situation, we encode processes in which conditions describe the structural and situational requirements and actions describe its function. There are twelve tool-related processes and we can divide them into two kinds. The first kind needs a special spatial situation (eg. An alcohol lamp does heating function when it is on and it is located under a tool containing reagents within a certain distance.). This is similar to yes-function-in-structure in [16]. The second does not need such structural requirement. It just has certain conditions concerning its own attributes. An example is that a hose moves an input liquid reagent from inlet to outlet when the pressure of inlet is greater than that of outlet.

Except a drying tower and a liebigh condenser belonging to the Connector type, readers can see common properties of each tool type in Table 2. We could discuss how a liebigh condenser works at the low level associated with water flow and heat transfer, and could make ACL do the same way of reasoning. However, as far as a liebigh condenser is concerned, it is enough in ACL to have only functional knowledge at an abstract level and to ignore all the details of the low level.

pLiebigh_Condensing:

```
(precondidtion) LIEBIGH_CONDENSER is used.
(condition)
GET_IO1_REAGENT(x,LIEBIGH_CONDENSER) &
GET_IO2_REAGENT(y,LIEBIGH_CONDENSER) &
GREATER_THAN(PRESSURE(x),PRESSURE(y))
(action)
MOVE_REAGENT_FROM_IO1_TO_IO2(x,x',
                DECIDE_RATE_BY_P_DIFF(x,y)) &
SET_TEMPERTURE(x', 25) /* decide the proper
state for x' */
```

pAlcohol_Heat_Transfer :

```
(precondidtion) ALCOHOL_LAMP is used as a
tool component.
(condition)
IS_ON(ALCOHOL_LAMP) &
GET_ABOVE_TOOL(x, ALCOHOL_LAMP) &
IS_CONTAINER(x) &
GET_AT_BOTTOM_REAGENT(y, x) &
LESS_THAN (TEMPERATURE(y),
                TEMPERATURE(ALCOHOL_LAMP))
/* where the temperature of the alcohol lamp is
set to be 800℃ in ACL */
(action)
CHANGE_TEMP(y, DECIDE_RATE_BY_DIST(x,
                ALCHOHOL_LAMP))
/* where the second argument is a factor to
decide the change rate of temperature */
```

4.2.2 Processes Related to Reagents

Reagents can have their own processes, and pseudo codes of two process examples related to reagents are shown below.

pMix-Liquid :

```
(precondition) Liquid reagents are used.
(condition)
IS_LIQUID(x) & IS_LIQUID(y) &
CONTACT_REAGENT(x, y) &
EQUAL(POLARITY(x), POLARITY(y))
(action)
GENERATE_HOMOMIXTURE(x, y)
```

pMwithH2O :

```
(precondition) An inorganic metal compound
reagent (x) is used.
(condition)
IS_ALKALINE(x) & IS_SOLID(x) &
CONTACT(x, y) & IS_H2O_LIQUID(y)
(action)
CHEMICAL_REACTION(x, {x+}, y, {H2, 2OH-},
                VERY-QUICKLY)
```

4.2.3 Physical natural law

There are several natural laws which do not belong to any class, and are applied to every experimental component. They are mainly related to physics. The heat-transfer process is shown as an example.

pHeatTransfer:

```
(precondition) Two reagents are used.
CONTACT(x, y) &
NOT(EQUAL(TEMPERATURE(x),
        TEMPERATURE(y))
(action)
  HEAT_TRANSFER_FROM_HIGH_TO_LOW(x,y,
        DECIDE_RATE_BY_T_DIFF(x, y))
```

4.3 Inference Engine

Inferencing in ACL occurs in two steps. The first, as mentioned above, is that only processes related to currently using

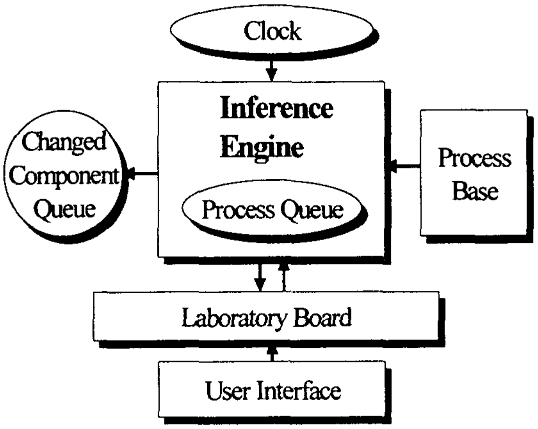


Figure 3. The ACL Inference Engine and Related Modules

experimental components among many become candidates, and this is done by the object-oriented environment. The second is that, once candidate processes are selected, the inference engine checks the conditions of each process and decides which processes take actions. Figure 3 shows the structure of the ACL inference engine and related modules. The clock makes signals at a constant interval which is adjustable, and the engine does inferencing at each clock. The process base includes all processes discussed in the previous sections. The CC (Changed Component) queue is a list including tools and reagents which are new, or change the value(s) of one or more attributes at the previous clock. The PA (Process-Action) queue is a list of actions such that the condition of the process has been satisfied and there has been no change which makes the condition false ever since satisfied. Using the CC queue and the PA queue makes ACL efficient in terms of inferencing speed. We discuss more in detail how the engine works.

All participating experimental components in an experiment are actually saved and maintained in LB consisting of ToolLabBoard for tools and ReagentLabBoard for reagents. A tool component in ToolLabBoard has relevant processes, attribute values such as kind, name, volume, position and orientation at the interface window, and spatial relations with other tools and reagents. A reagent component in ReagentLabBoard also has related processes, attribute values, and spatial relations with other reagents and tools. The inference engine examines processes related to each laboratory board component of LB and checks which processes are satisfied to be activated. When there exist satisfied processes, an experiment progresses by invoking the process actions, and then the situation is changed.

For speed efficiency, ACL needs to have the capability to ignore temporarily experimental components that do not affect the progress of experiments. For instance, once processes related to a reagent are not satisfied to be activated at a clock t_i and there is no change of the reagent (i.e. no change of an attribute value, spatial relation, or status) between t_i and t_{i+1}, the processes should not be considered as processing candidates at t_{i+1}. This method is called eliminating temporal redundancy. We use the CC queue for this purpose. New components of an experiment are also inserted into the CC queue at the beginning. The inference engine checks relevant processes of tools and each reagent in the CC queue at each clock, and examines whether the conditions of the processes are satisfied.

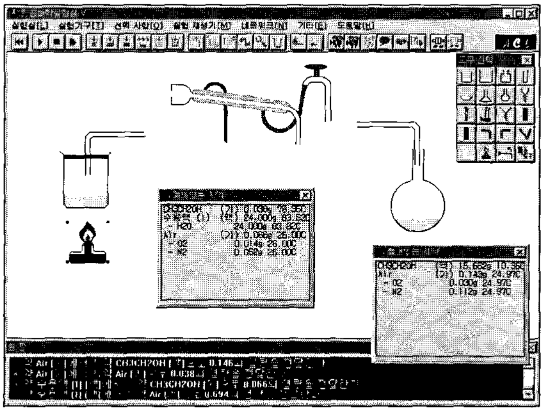
One more important feature of our system is using the PA queue. When an alcohol lamp is on and there are a beaker above the lamp and a reagent in the beaker, the pAlcohol_Heat_Transfer process works and increases the temperature of the reagent. If there is no change of configuration, the process must work continuously. In this case, the action of the process is inserted into the PA queue and the tool component is eliminated from the CC queue. When there is a change like turning off the lamp, the action is eliminated from the PA queue and the tool component is inserted again in the CC queue. The action part of a process satisfied by the processes related to tools is invoked periodically until there is a change of the corresponding component. Because there is no need of unnecessary checking of many processes, the engine has speed efficiency.

Since acting processes are at different speed, the relative speed is represented in our work by fuzzy speed VERY FAST, FAST, NORMAL, SLOW, VERY SLOW.

5. Implementation and an Experiment Example

The ACL system consists of a knowledge base, an inference engine, a visualization module, an explanation module, a user interface module, and a laboratory board in which an internal cognitive model is maintained. Using Borland C++ version 4.52, we have developed ACL running on PC with MS Windows 95, and network operations are possible in TCP/IP network environments. We show one experiment example in this section.

The example is the separation distillation experiment in which the liquid having the lower boiling point is distilled from liquid mixture by heating. Because ethanol has lower boiling point than acetic acid, the mixture is distilled by the difference of boiling point. Figure 4 shows the window copy of this experiment. Readers can see various menus and tools on the window.



```
CH3CH2OH      (기) 0.030g 78.35C
수용액 (1)    (액) 24.000g 83.82C
 - H2O            24.000g 83.82C
Air           (기) 0.066g 25.00C
 - O2             0.014g 25.00C
 - N2             0.052g 25.00C
```

```
CH3CH2OH      (액) 15.662g 10.35C
Air           (기) 0.142g 24.97C
 - O2             0.030g 24.97C
 - N2             0.112g 24.97C
```

Figure 4. Separation Distillation Experiment

This experiment is progressed as below.

① Ethanol is mixed with acetic acid by process pMixLiquid that mixes two reagents if the two has the same polarity. Liquid homogeneous mixture is created by this process.

② If the alcohol lamp is turned on by user action, heat is transferred to beaker by process pHeating. And then liquid homogeneous mixture and its components has the higher temperature.

③ Ethanol is reached to the boiling point before acetic acid by continuous heating. Liquid ethanol begins to change to gas ethanol by process pVaporizing.

④ During gas ethanol is created, internal pressure is increased and gas passes through glass tube by process pFlowReagent. The gas temperature in Liebigh Condenser is decreased and the gas changes to liquid. That liquid is gathered in the flask.

6. Summary

We believe that science experiments in a laboratory are essential for science education. However, students do not have enough experiment chances in reality. One of alternatives to solve this problem is to build a science-experiment system on PC and to have students and teachers use the system and do any experiments on computers.

In order to build such a program, we suggest the representation of a 2-dimentional model and ontological entities necessary in causal reasoning of scientic experiments, and the inferencing method to derive system behavior.

For the ontological entities of chemistry experiments, we consider experimental tools, reagents and their heirarchical structures, physics and chemistry natural laws, and functional abstraction knowledge. We have developed ACL which provides an experiment environment where a student can do any experiment (i.e. non-predetermined experiment) as far as related information and knowledge are provided, and shows experimental system behavior similar to what happens in the same situation in a real world and descriptions about why it happens. The use of ACL shows the usefulness of suggested methods.

The knowledge base in ACL has a hierarchical structure and consists of two major parts which are the Object class for tangible things and the Concept class for intangible things. Each class has its own subclasses. LB maintains an internal model which represents experimental situations corresponding to experimental behavior. The spatial information between tools and reagents are saved in LB.

Inferencing in ACL occurs in two steps. The first is that only processes related to currently using experimental components among many become candidates, and the second is that once candidate processes are selected, the engine checks the conditions of each process and decides which processes take actions. The engine uses a strategy removing temporal redundancy for efficiency.

The user interface is provided to allow students to do experimental behavior, and to show experimental phenomena. Explanations about behavior are also provided to users so that they understand why such things happen.

References

[1] D.G. Bobrow, "Qualitative Reasoning about Physical Systems", MIT Press, 1985.

[2] P. Bose and S.A. Rajamoney, "Compositional Model-Based Design", the 13th IJCAI, 1993.

[3] http://www.sfu.ca/chemed/; Instructional software for chemistry; electronic textbook

[4] R. Davis and W. Hamscher, "Model-based Reasoning : troubleshooting", In E. Shrobe(Ed.), Exploring Artificial Intelligence, Los Altos, CA : Morgan Kaufman, 1988.

[5] P.D. Gautier and T.R. Gruber, "Generating Explanations of Device Behavior Using Compositional Modeling and Causal Ordering", AAAI, 1993.

[6] P. Hayes, "Naive Physics I: Ontology for Liquids", In J.R.Hobbs and R.C.Moore (Eds), Formal Theories of the Commonsense World, ABLEX Publishing Co., 1985

[7] Y. Iwasaki, "Qualitative Physics", AI handbook Vol.4 Chapter XXI, 1989.

[8] S.H.Koo "Qualitative Analysis of Particul Systems", Ph.D. Thesis, Dept. of CS, University of Southern California, 1994.

[9] B.Kuipers, "Qualitative Reasoning: Model and Simulation with Incomplete Knowledge", MIT Press, 1994

[10] A.M. Law and W.D. Kelton, "Simula Modeling and Analysis", McGraw-Hill Inc., 1982.

[11] D. Lenat and R.V. Guha, "building Lagrge Knowledge-Based Systems: Representation and Inference in the CYC Project", Addison-Wesley, 1990

[12] http://www.ornl.gov

[13] R. Reiter, "A Theory of Diagnosis from Principles", AI 32:57-95, 1987.

[14] http://links.math.rpi.edu/~markv/chemistry.html, 1998

[15] R. Shank, "AI, Multimedia, and Education", the 13th IJCAI, 1993.

[16] M. Vescovi, Y. Iwasaki and B. Chandrasekaran, "CFRL : A Language for Specifying the Causal Functionality of Engineered Device", AAAI, 1993.