Instruction Design by using Unified Modeling Language for E-learning

Keiji EMI (Kanazawa Institute of Technology)
Akira YAJIMA (Osaka International University)
Kikuo TANAKA (Osaka University)
Michio CHUJO (Kwansei Gakuin University)
Akira NAKAMURA (Crossroad, Math Navigator)

Many software developers have recently paid attention to an "object-orientated method". They are not only interested in object-oriented programming, but also in the class design in the system analysis. As for education or instruction as well, this viewpoint is indispensable to the instruction design that requires easy-to-understand teaching materials and students' essential understanding. In this paper, we suggest that the instruction design should be applied to UML (Unified Modeling Language) that is one of object-oriented method.

1. Introduction
1.1 Knowledge begins with classification

Including the case of mathematics or physics [1], learning or knowledge generally begin with the classification of the things. For example, some programming languages, such as Fortran, BASIC and C, are procedure-type languages. On the contrary, C++, JAVA and Visual Basic.NET are object-oriented programming languages. Classification leads to a concept of "class" in object-oriented design. The "class" described here in this paper means a set of concepts or things. It's a little different from the "class" of object-oriented programming. We propose that using the similar idea to object-oriented programming can be applied to instruction design. First, we guess, one obstacle to learners understanding a new knowledge and accustomed its usage is the lack of concrete examples. As for the above-mentioned examples, even if Fortran, BASIC, C, C++, JAVA and Visual Basic.NET are classified and taught as procedure-type languages or object-oriented one to those who have never seen or touch such programming languages.

Second, even if learners understand the classification, they often confuse which rank the class exists on. By applying the knowledge repeatedly, learners can get rid of the confusion. It is important to teach learners that classes exist on ranks or in the "layered structure".

The systematic classification of biology, an independent name is given to each rank or each layer; kingdom, division or phylum, class, order, family, genus, species. In other words, relations between the super class (an upper class) and the sub-class (a lower class) are clear.

Therefore, if we use an object-oriented design for instruction, a technical term or terminology can be treated as the class name. Then, we guess we should treat real instances as concrete instances.

1.2 Object-oriented Modeling

The "object-oriented modeling and design" is thought very useful in order to understand the requirements, to clarify the design, and to easily maintain the system.

When it comes to an object-oriented modeling, UML is a diagram that doesn't depend on the specific object-oriented programming language. It is thought that a real world is modeled by "object", and that it is designed by using the object-oriented model.

In the industrial world of information technology, the analysis of business in enterprises by UML is getting more and more popular. In Japan, when a UML seminar is held, it is full of attendants. The attention to UML is
getting stronger. In enterprises, it is understood that “the
object of business” is equal to “the flow of the object”.

Until UML appeared, there have been some
methodologies; Code&Yourdon’s methodology [2],
Shlaer&Mellor methodology, Booch’s methodology [3]
and OMT (Object Modeling Technique) [4]. Booch’s
methodology and OMT are unified to UML.

UML consists of three kinds of models: the object
model, the dynamic model, and the function model. In
this paper, we deal only with the object model, because
we just analyze lecture plans or the development of
teaching materials with UML.

2. Object model description and its
elementary concept

For example, we show the instruction design or
teaching materials to explain the structure of PCs. Three
kinds of relations are explained by using this figure.

![Diagram of PC parts](image)

Figure 1. The structure of a personal computer.

2.1 Aggregation relation (“part-of relation”)

A PC (personal computer) consists of a body, storage
devices, a memory, a CPU, a mainboard, a mouse and a
keyboard. A super class (in this case, “PC” class)
consists of all sub-classes. In this case, all sub-classes
must be present. When this is applied to the
development of teaching materials, you can make sub-
classes independently, but you have to make materials
of all sub-classes. In this case, when you explain the
structure of PCs, you are apt to skip the explanation of a
mouse and a keyboard, because they are naturally
present. Even if you may skip the explanation of a
mouse and a keyboard in your lecture to save time, you
have to mention that they are necessary parts of a PC.

2.2 Generalization relation (“is-a relation”)

Next, various classes exist as a sub-class in the
storage device class in the FDD class, the HDD class,
the CD-ROM Drive class, and so on. The storage
device class has two characteristics. First, tracing from
a super-class node, there are some sub-classes. The
number of sub-classes is not limited. A sub-class is an
example or an instance of the super-class. Second, a
sub-class inherits the characteristics of its super-class. It
is so-called “inheritance relation”.

Relations between such a sub-class and the super
class are called “generalization relation”, “is-a relation”,
or “inheritance relation”. From the viewpoint of
development of teaching materials, all you have to do is
to develop a sub-class after the teaching materials of the
super-class are developed.

When you make teaching materials about a storage
device, you have to take out the characteristics that a
FDD, a HDD and a CD-ROM drive commonly have.
After that, you may make teaching materials about a
FDD, and deal with the characteristics that a FDD only
has.

2.3 Instantiation relation.

Each class always has instances. For example, the CD-
ROM drive class has a R-2432X of the T Incorporation
or a RW-1664E of the I Incorporation. When you give a
lecture, you should show students product photos or
catalogs.

3. A good explanation and a bad
explanation

3.1 Classifications of printers
In the beginning of 1990s, the classification of printers for a personal computer was in the following in figure 2. This classification might be only in Japan.

At that time, a personal computer was installed with a character-based operating system, therefore the main stream of printers was a line printer which printed out line by line.

A laser printer was not called a "laser printer", but a "page printer". It was called so, in order to emphasize that a laser printer can print out page by page. There are double standards for the classification. One is a kind of its ink, and the other is printing unit (line or page).

**Classification of printers**
(a bad example)

```
  printer
  /  \
Impact Thermal Ink Jet page
```

Figure 2. The bad example of the classification of printers. It isn't being classified in the kind of the printing ink or the printing unit.

**Classification of printers**
(a good example)

```
  printer
  /  \/
Impact Thermal Ink Jet Laser
  /  \
Line page
```

Figure 3. The good example of the classification of printers. There are double standards of the classification.

Because of the emphasis in the description of important terms, the word of "the line printer" may be omitted in the description of the textbook.

But, in order to be understood the important terms logically, you have to make a teaching material after you decide how to classify the important terms.

We guess that it is easy to give a lecture or make a teaching material after making a robust instruction design.

**3.2 Don’t skip the opposite concept to a concept.**

When terminology is classified with UML, you will often feel that the rank structure is not so suitable. Think about the data type in the programming language, for instance. The data type has the numerical value type and the character type. And the numerical value type usually has the integer type and the floating point number type.

Then is there a fixed point number type?
The answer is yes. The fixed point number type is used in JAVA language running on the cellular phone programs. Fixed point numbers are used for the effective performance of the CPU. The fixed point number type is only used in the special case, so it is usually skipped. Still, you will have to deal with both the "floating point number type" and the "fixed point number type".

Talking about "analog", next you have to deal with "digital". Talking about "general", next you have to deal with "special". A distinction is necessary to form a certain concept.

We think that it is effective to use the pair of the opposite concepts (so called "discriminator"). When you can perfectly divide a super class into two with the opposite concepts, it is called "complete". On the other hand, when you cannot perfectly divide a super class into two with the opposite concepts, it is called "incomplete" [5]. If it is complete, the inheritance is very important.

**4. Application to mathematics problems**

The idea of classifying teaching items by hierarchical relations and defining teaching strategy on them is not new. So let’s see where and how object-oriented method is applied, and benefits of applying object-
oriented approach on teaching material analysis in the case of mathematics.

It is possible to make web pages on mathematical formulae and their combinations with UML. You can regard example problems and exercises in which a mathematical formula can be applied as instances of the formula classes. If you can open the formula classes and close the instances, you may be in business on e-learning.

We suggested the following plan. As for the universal knowledge such as mathematical formulae, they should be free and open. On the contrary, for both the combination of the universal knowledge and the instances, a fee should be charged. Thinking about such business in the future on Math Navigator (a web site)[6], all the mathematical formulae will be free and open. On this site, mathematical formulae (class) will be linked to the explanation pages of entrance examination problems (instance).

When students poor at mathematics use this site, they need more instances than good students. When students good at mathematics use it, they can use more hierarchies than poor students.

Figure 4. Math Navigator. Now only in Japanese.

5 Discussions and summary

We can tell about two statements with object modeling. The former is equivalent to tell the significance of the upper class. In other word, although subclasses exist and inheritance relations are present, the subclasses can be ignored. Students cannot develop their ability to solve concrete problems. On the other hand, the latter one is that students know only instances, but not the theory (or super class).

Students cannot apply their ability to solve other problems, because they do not know the theory or the super class.

One subclass happens to inherit form more than two classes. It is called "multiple inheritances [7]", as for the multiple inheritances, it isn't being taken into consideration in earnest in this paper.

We do not deal with the details about hierarchical relations of sub-classes and instantiation relations, but we will make teaching materials and evaluate them later.

References


This site is written only in Japanese.


http://www.crossroad.jp/

The entrance examination is in the following pages.

http://www.crossroad.jp/mathnavi/math-i/math-i-index.html