The Manifesto of Lyee

Lyee's methodology for software development has already made an outstanding performance and a revolutionary effect on the actual business field. The developer's original idea, which nobody has been able to put into practice, and the theory and methodology deduced from his idea are aimed to grasp what is software from viewpoints of consciousness, existence, and meaning. If you take an ordinarily favorable stance to look at the developer's theory, you may first feel that this methodology looks very similar to the object-oriented approach, OOA. If you take a closer look at it, however, you will soon realize that your first impression is only ungrounded. This is because you will learn that Lyee's methodology excludes most of the claims that the data-oriented approach, DOA, makes and Lyee's theory clarifies the principle of software.

Based on Lyee's principle and taking a perspective of scientific history to look into the current situation of the field, you will soon know that the existing software principle and methods have never and will never strike a point of software so that those will be surely replaced by Lyee.

We are certain that Lyee's theory will give numerous important suggestions to software related people such as researchers, scholars, system engineers and whoever, and also urge them to reflect the current situation of the software world.

A purpose of Lyee

The purpose of this research can be summarized to seek for a method to establish, by determinism, the relationship of a software object model and a program model.
In other words, it may be regarded as a study on the software development methodology.

Related positioning, features and procedures are illustrated in Fig.1.

As essential information to understand the logic of Lyee, the attitude on this research is explained below.

Looking back the last 20 years or so, it looks that the trend of software research has

buried into oblivion as separate problems such problems as the instability of a software object model and a program model which is determined by self-will.

This means that, notwithstanding that problems of this kind are root subjects of software, they have been abandoned for various reasons as problems impossible to solve.
Instead, alternate thinkings have become popular, thereby promoting researches on the technique of the formality which is fractionalized for various fields.

And, the trend has eventually invited an era missing the essence of software, thereby leaving problems to the future, similar to the problem of pollution by industrial waste.

This study has made direct research on this exact fundamental subject of software.
Therefore, the solution to the above two problems becomes its central subject.
Based this research's attitude, a practical method for materializing a thinking method to avoid and conquer the above problems has been introduced, which is not, for example, a model designing theory based on knowledge.

Concluded.