"ALGEBRA OF SYSTEMS: A METALANGUAGE FOR MODEL SYNTHESIS AND EVALUATION"

This paper is different in two aspects: first is that it was written by Professor Koo, so we are able to obtain a direct access to the field which our instructor works on; the second is that it focuses more on mathematical theory and simulation of the process of designing, something that the previous papers about the concept of database rarely mention. I learn much from reading this paper, including the algebraic theory, technique for simulation, and the application of the methodology.

1. Algebra of Systems

The algebra discussed here is different from the linear algebra we learned in the freshman year and the relational algebra we learned from E.F.Codd. As far as I'm concerned, the algebra here is a more general idea that other forms of algebras can generate from it with additional constrains and information.

The two vital things in algebra are operand set and operator set, and as long as arbitrary compositions of operators on elements in operands result in elements in operands. It is easy to understand using Euclidean space. The operands in a Euclidean space are vectors, and the operators include addition and inner product.

It won't take us long to figure out that the result of addition or inner product is another vector in the same Euclidean space. Think about it a little more, and we find that this is just the foundation of linear algebra.

Algebra of systems has its own operands and operators as the definition of algebra requires. The operands can be other domains, and this property is called many-sorted algebra, which provides great efficiency in building up the structure of AoS. Since AoS is developed to solve the real design problems, its operands and operators must closely related to the practical issues, therefore, there are three levels of operands, denoted in P, B, and C in the paper.

P refers to the properties domain, which contains the quantitative and qualitative properties of a real problem. As a result of arithmetic calculation in the field of algebra, the returned value is of great significance for evaluation, and the property of properties domain enables the calculation and comparison of these values.

B refers to the Boolean domain, which is essential in judging the result generated by the properties domain.C refers to the composition domain with a bipartite graph data structure, meaning that there are only two kinds of element in the composition domain, namely, object and processes. This property makes it easier to utilize computational models designed for solve bipartite graphs and thus accelerate the speed of calculation. I think the composition domain is the most important one in the three domains of AoS since it establishes a connection between the algebra space and the real world. This is essential for a practical algorithm or software, and should be taken seriously especially for industrial engineers.

Notice that the formal syntax of P when treating it as a textual language can be specified in the EBNF, which is the abbreviation of Extended Backus Naur Form, a metasyntax to express context-free grammars. Therefore, it is a regulated computer language. In my opinion, this property has a direct relation to the successful application of OPN.

2. Design Process Abstraction, Simulation, and Future

The main idea of AoS can be illustrated by the figure in the paper clearly. My understanding of this process is as follows: the methodology itself combines the extant knowledge (enumerable model) to generate many candidate solutions (generated sub-models), then evaluating these solutions, choosing the desired ones and return them to the available knowledge. Do the iteration until certain constrains are satisfied, and choose the most beneficial plan.

Someone has mentioned that innovation is not the generation of new knowledge, but the successful combination of old knowledge. AoS just provided the method of generating combinations of old knowledge and evaluating these combinations. This work are traditionally done manually, thus the OPN-IDE is a really swift access to the models for designers. Also, it greatly reduces the cost for accomplishing this task by people.

One crucial point in successful applying the AoS is to properly simulate the real system so that it can fit the structure of AoS. More specifically, one has to determine the counterpart of objects and processes of composition domain in the real system before using OPN-IDE. This requires that the designers must have some basic knowledge about the AoS and its domains. Maybe the future development of OPN-IDE can eliminate such requirement so that even the system designer knows nothing about algebra he can use it without any trouble.

Finally, with increment of information in future systems, more powerful algorithms must be applied to the OPN-IDE so that faster access to the knowledge and suggestions can be obtained. I believe it is a program with a strong vitality and the future development of system will depend more on it.

Source: http://toyhouse.cc/profiles/blogs/personal-report-of-week-8