A Proposal for Intention Engineering

Fumio Negoro
The Institute of Computer Based Software Methodology and Technology, Tokyo, Japan
f-negoro@lyee.co.jp
and
Issam A. Hamid
Iwate Prefectural University, Iwate, Japan
Issam@soft.iwate-pu.ac.jp

Abstract

Discussed herein is to introduce that an engineering world of software can be defined based on a non-scientific hypothesis. In the present software engineering world, there is a trend that anything can be done in the expansion of applications and the improvement of functionality, but at the same time it has produced a variety of problems that cannot be overlooked, viewed from the social aspect. In other words, a cause can be found where the determinism of software could not find universality yet due to the pre-matured levels of basic engineering and basic science of software. This research was carried out with a viewpoint focusing on thereabouts. Speaking of the nature of software, it should be based on notions of philosophy, psychology, and ontology rather than so-called mathematical and engineering notions that seem to have played a central role in the software world, if viewed from its inherent immanence. However, due to its intense demand and application, the actualization of practicality was more emphasized. As a result, almost none of its essential problems has been improved over the last half century. In the near future, time will come when it is reviewed fundamentally as a problem of a social nature. This study provides an observation, from a viewpoint of ontology based on hypotheses that have been deduced from this study, to the nature of software from a relation between an intention and programs concerned with software.

1. Ontology in this study

In this study, Self means Existence that cannot dwell in Self. That is, Self is non-existent within the Self. It is existence other than Self that dwells in Self, and the existence is called Others. In this study, Self means a substance and Others mean attributes. Others exist really in the sense of dwelling in the Self. For example, notions of the spirit, mind and the whole correspond to Self. All things in nature, words, writings, pictures, programs, machines et al correspond to Others.

This study aims at obtaining a method that enables to define a relation of Self and Others from the Self standpoint. But it is considered impossible because Self needs to stand on a viewpoint of Others and it is impossible for Self to become Others. However, we cannot reach a stage to publicly understand Requirement unless the mechanism to make this possible is discovered. This method needs to be obtained by using axiomatic system. But, this subject is too essential to discuss in general, so the observation of this study is limited to the software world. That is, this viewpoint in the software world is to obtain theorems that rule a thinking action which can replace a composition in natural language (Requirement) into a program in programming language. Therefore, this standpoint itself is inevitably the standpoint of ontology. And, discussing this ontology means discussing a software development methodology. Ontology is basically a world based on a hypothesis. A hypothesis is the inevitable tool for discussing ontology. With ontology standpoint, the right or wrong shall be determined by the proficiency of hypothesis. Therefore, ontology in this study also takes the same viewpoint. This study is carried out by using axiomatic system. The hypothesis is gigantic, and all of it cannot be detailed thoroughly in this paper, but it is discussed herein as much as possible.

Theorems established in this hypothetical world are made into a function as a result. It is called Scenario Function (SF). SF is a set of propositions that take nouns belonging to the sentence with a role of a Requirement as Variable. Thus, to define SF is nothing but to define these propositions. The cause for materializing the sentence is Intention (Self). When SF is executed on a computer, Intention is established connotatively (functionally) by an

autonomous complimentary action of propositions. We cannot recognize the Intention. However, instead of it, Requirement (Others) that reflects the Intention materializes in the memory area. Then, it becomes possible to recognize the Requirement. In this connection, if the number of propositions defined by SF is designated by N, the number of possible Requirements obtained by the complementary action generated by executing SF on a computer becomes N^N at a maximum. A proposition cited herein is an action that attempts to actualize Self to become Others.

2. Logic of discovery

For example, meaning of a sentence "I know an apple" is always uncertain if "I" is Self, because Self cannot become Others. But if Self can become Others, it will become certain. If we can also reside in another world at the same time and if we can discuss the Self world from thereabouts, the situation of inference will change. However, such relation is illusion. To overcome such a problem, a hypothetical world that enables it axiomatically is constructed. This hypothetical world is called Consciousness Model (CSM), SF, and Predicate Structure. By using the structure of SF, the Self can become Others. SF and Predicate Structure are established by theorems of CSM. SF and Predicate Structure can be defined by any programming language. A structure that is a base for SF is deduced by theorems of CSM. This structure is called Three-Dimension-like Space Model (TDM). Details of CSM, TDM, SF, and Predicate Structure are discussed later.

3. Consciousness model

With this study, it is considered that time corresponds to the speed of light. It hypothesizes that time could be faster or slower than the speed of light and there are particles to represent them. A countless number of such particles exist and they are independent of each other. The speed of light is called Time Speed. Characteristics of such particles are determined individually by characteristics of unique rational number (UR) which we cannot cognize. Details of UR are not discussed further herein. These particles are called Logical Atom. An image of Logical Atom is discussed herein. Our image of characteristic of Logical Atom is that it exists if we think it exists and it does not exist if we do not think it exists. In our world, a similar thing to Logical Atom is words. Words can become a sentence and a language. In other words, Logical Atom is hidden in a sentence and a language. Thus, a sentence and a language can be considered as objectified Logical Atom. However, we cannot cognize Logical Atom itself. We consider that the speed of the Existence is faster than the speed of light and that is why we cannot cognize it. CSM is defined by using Logical Atom. The aim of creating CSM is to establish axiomatic system to materialize and objectify Intention.

3.1. Unknown space

The original set of Logical Atoms is called Unknown Space (UKS). Logical Atoms of the same Time Speed cannot exist in UKS. Therefore, if duplication occurs, one of them will shift to outside of the UKS. This paper does not discuss the duplication in detail. A space outside this UKS is called Ideal Space. The order of the shifting Logical Atoms is unrelated to the values of Time Speed. However, intervals of the shifting Logical Atoms reflect the value of Time Speed of the preceding Logical Atom.
Parameters established in UKS are defined as follows. Parameters defined herein belong to UR and exist in UKS.

- Time Speed of UKS: $\Sigma_1^{\phi} V_i$
- Idea of space of UKS: $\varepsilon \equiv 1/2\Sigma_1^{\phi} V_i$

ε is the first number of rational number which we cannot cognize.

- Ψ is the total number of Logical Atoms in UKS

Ψ is a natural number which we cannot cognize.

- $\Psi = 2\ \Sigma_1^{\phi} V_i$

3.2. Ideal space

When Logical Atom existent in Ideal Space (IDS) becomes more than 3 pieces, their Time Speed is arranged from large to small, and the Logical Atom lying at the central ranking place is called Boundary Atom. When the total number of Logical Atoms in IDS is an even number, one out of two in central ranking place with smaller Time Speed becomes Boundary Atom. Time Speed of Boundary Atom is expressed by V_B. Boundary Atom is re-defined every time one of Logical Atoms is shifted from UKS. Parameters established in IDS are defined as follows. Parameters defined herein belong to UR and exist in IDS.

- V_i is Time Speed of Logical Atoms that exist in IDS.
- V_i in the relation of $(V_i - V_B) > 0$ is called Consciousness Atom (CNA) and it is expressed by V^+_i.
- V_i in the relation of $(V_i - V_B) < 0$ is called Cognitive Atom (CGA) and it is expressed by V^-_i.
- $| V_i - V_B |$ is called relative difference (RD) of V_i.

Once Boundary Atom is determined, the coordinates of IDS are determined.

- V_B becomes the coordinates of UKS.
- The coordinates of V_B are $(V_B + \varepsilon)$.
- All CNA take Boundary Atom's coordinates as their coordinates.
- All CGA take their own Time Speed as their

respective coordinates.

- Inverse numbers of RD are called Occupied Space. Occupied Space is an idea of space. Occupied Space of CNA is expressed by M^+_i and Occupied Space of CGA is expressed by M^-_i.

If parameters herein do not exist in IDS as Logical Atoms, they cannot be established. If Logical Atoms cannot establish all the above-mentioned parameters, they are not allowed to have the coordinates. However, the coordinates of Boundary Atom and UKS can always be established. Logical Atoms whose coordinates cannot be established exist in IDS but they cannot be used as factors of the deduction in CSM.

- The Occupied Space of Boundary Atom is expressed by $2\ \Sigma_1^{\phi} V_i\ (= \Psi)$.

It is the largest Occupied Space among Logical Atoms. Therefore, the Occupied Space represents IDS.

3.3. Subset

A set comprising of CNA and a set comprising of CGA are created every time new Logical Atom appears from UKS. That is, IDS comprises of two sets. And then, among each set of CNA and CGA, respective subsets are created. The subset of CNA is expressed by S^+_i and the subset of CGA is expressed by S^-_i. If the number of CNA in IDS is designated by p, the number of S^+_i is expressed by the following formula. $pC3+pC9+ \cdots +pC3^{\lambda}$, and hereupon, C is a combination symbol, 3^{λ} is the number of CNA belonging to S^+_i, and λ is a natural number. More precisely, S^+_i is expressed by $S^+(\lambda, i)$. S^-_i is created in the same manner as S^+_i and expressed more precisely by $S^-(\lambda, i)$. S^+_i and S^-_i are created in the order from small to large values of λ. In this case, plural numbers of subsets are created for the same value of λ. The order of creation in this case is discussed later.

3.4. Significant space

CNA belonging to $S^+(\lambda, i)$ are arranged from large to small of Time Speed and the sum of M^+_i located at its even number positions is called Significant Space of $S^+(\lambda, i)$. It is expressed by $M^+(\lambda, i)$. "i" of M^+_i and "i" of $M^+(\lambda, i)$ are different. That is, "i" of M^+_i is an identifier of CNA and "i" of $M^+(\lambda, i)$ is the order of creation. CGA belonging to $S^-(\lambda, j)$ are arranged from small to large of Time Speed and the sum of M^-_j located at its even number positions is called Significant Space of $S^-(\lambda, j)$. It is expressed by $M^-(\lambda, j)$. "j" of M^-_j and "j" of $M^-(\lambda, j)$ are different. That is, "j" of M^-_j is an identifier of CNA and "j" of $M^-(\lambda, j)$ is the order of creation. The reason of the arrangement order of Logical Atom is not discussed herein.

3.5. Unit

In IDS, a theorem can be established which is related to the smallest difference between Significant Space and Occupied Space. A relation between Significant Space and Occupied Space that materializes the smallest difference is called Unit. In this case, Logical Atom that materializes Occupied Space is particularly called Representative Atom (RPA).

Two types of Unit are defined as follows:

- A relation of Occupied Space that is approximate to but smaller than $M^+(\lambda, i)$ and $M^+(\lambda, i)$ is called Consciousness Unit (CNU). In this case, CNA to represent the Occupied Space should belong to $S^+(\lambda, i)$.
- A relation of Occupied Space that is approximate to but larger than $M^-(\lambda, j)$ and $M^-(\lambda, j)$ is called Cognitive Unit (CGU). In this case, CGA to represent the Occupied Space should not belong to $S^-(\lambda, j)$.

$S^-(\lambda, j)$ and CGA to represent Occupied Space correspond to Self. That is, this means a substance of Existence. CGA belonging to $S^-(\lambda, j)$ corresponds to Attributes of the Existence. Hereinafter, CGA belonging to $S^-(\lambda, j)$ is referred to as Attributes.

The above-mentioned corresponding relation can also be established for CGU. As a result, the following can be said. CGU represents a relation in which Self does not belong to itself, and CNU represents a relation in which Self belongs to itself. The number of subsets becomes larger than that of Logical Atoms. Therefore, it does not mean that all subsets can have Representative Atom. According to the theorems established in CSM, one Logical Atom cannot become Representative Atom of a different Unit by nature. From this reason, Unit is created as many numbers as Representative Atoms are established. In other words, the number is the same as that of Logical Atoms existing in IDS. Subsets play a role of a pattern that gives characteristics to IDS and determine Representative Atom that determines Unit. Unit becomes something that materializes Self. CNU represents a relation in which Self grasps itself. Therefore, establishing a relation that makes CGU correspond to CNU means that Self in CGU is grasped by CNU. In other words, it means that non-existent Self can become existent. Details are discussed later.

4. Cognitive unit (CGU)

All CNU are determined by only one theorem. In contrast, CGU is determined by five kinds of theorems. Details of the theorem to determine CNU are not discussed, but outlines of the five kinds of theorems of CGU are explained herein.

4.1. Explanation of CGU

4.1.1. Probabilistic unit (PBU)

The first theorem establishes CGU based on probability. This is called Probabilistic Unit (PBU) and this Unit is defined as follows. The number of $S^{-}(\lambda, j)$ that materialize in IDS is expressed by N. Firstly, duplex permutations (N^{n}) are created from this N. Secondly, one permutation is selected from them in an arbitrary manner. Thirdly, plural numbers of $S^{-}(\lambda, j)$ belonging to the selected permutation are arranged by a sequence column in an arbitrary manner. Finally, Representative Atom is determined for $S^{-}(\lambda, j)$ that is placed at the head of the sequence column.

4.1.2. Normalization unit (NMU)

The second theorem establishes CGU based on Association, which is a relation established between one PBU and all CNU. This is called Normalization Unit (NMU). $M^{+}{}_{i}$ of RPA of CNU whose size is close to that of $M^{-}{}_{j}$ of RPA of PBU but smaller is selected. CGA whose size is close to that of the $M^{+}{}_{i}$ but larger is selected. In this case, the number of CNA belonging to $S^{+}(\lambda, i)$ of the CNU is expressed by n. $S^{-}(\lambda, j)$ that satisfy RPA of CGA are selected from the Permutation that established PBU. However, the number of CGA belonging to the selected $S^{-}(\lambda, j)$ should be the same as the number of CNA belonging to $S^{+}(\lambda, j)$ of the CNU.

4.13. Representative unit (RPU)

The third theorem establishes CGU based on multiplexed NMU. If the number of CGA belonging to $S^{-}(\lambda, j)$ of the NMU is expressed by m, in this case, new subsets are created from the $S^{-}(\lambda, j)$. The number of new subsets to be created is m^{m}. All CGA belonging to m^{m} of new subsets should belong to the $S^{-}(\lambda, j)$. RPA of the NMU is copied and becomes RPA of each of new subsets. Unit determined by this manner is called Representative Unit (RPU). Among these Units, the NMU is contained. This Unit is eliminated from these RPU. Furthermore, these Units that cannot satisfy the conditions of CGU are eliminated from these RPU.

4.1.4. RPU created by coupling

Among subsets that establish RPU, some of them have the same CGA as elements. And, there is also the same RPA. Subsets that establish CNU, PBU, and NMU are defined in IDS. Subsets that establish RPU and NTU (discussed later) are different from subsets that establish CNU, PBU, and NMU. As mentioned earlier, there is a relation where a subset that establishes RPU is created from a subset that establishes NMU. And, there is a relation where a subset that establishes NTU is created from a subset that establishes RPU. Under the theorem in CSM, the same CGA should not belong to one subset. However, the same CGA exist in a subset that establishes RPU. This is against the theorem. In order to avoid this contradiction, the fourth theorem is derived. The fourth theorem has the following action. This action is called Coupling.

It produces one new unit from a pair of the existing RPU. This new unit is also RPU in terms of its characteristics. However, RPU that is against the theorem shall not be eliminated by Coupling. Therefore, RPU that is against the theorem and RPU created by Coupling exist together in Representative Space (RPS), which is explained in the following sections. Regardless of the action of the fourth theorem, contradiction still exits in RPS. Only a fact that RPU increases is established. This Coupling does not continue forever because there are certain conditions to establish Coupling and it stops when the conditions are not met. Details of the conditions are not discussed herein. [1]

4.1.5. Natural unit (NTU)

The nature of some RPU changes in a process of Coupling. Details of this change are not discussed herein. The RPU whose nature has changed is called Natural Unit (NTU). The fifth theorem is new Coupling concerned with NTU. Units created by this new Coupling are also NTU. But, this new Coupling shall never stop, unlike that of RPU. Details of the reason why it never stops are not discussed herein. Under the theorems established, the world that contains contradiction is created, which establishes our world.

4.1.6. Supplementary explanation of coupling

Units are created when new Atoms appear in IDS. Then, CNU and CGU are created. If the same Unit already exists, it will not created repeatedly. Coupling starts for a determined pair of NTU. This Coupling stops temporarily when new Atoms appear in IDS, and continues again including new NTU. Coupling history means a set of created Units between an interval when Coupling starts and when it stops temporarily. The size of Significant Space of a subset to establish NTU in Coupling history becomes larger as Coupling proceeds.

4.2. Supplementary explanation of unit

4.2.1. Location of existence of representative atom

The same Logical Atom cannot be used for Representative Atom (RPA) that establish CNU, RPA that establish PBU, and RPA that establish NMU. As mentioned earlier, the same Logical Atom has to be used as RPA that establish RPU and NTU as a result. Logical Atoms that establish subsets exist in IDS. As RPA are copies of Logical Atoms that exist in IDS, RPA shift to the outside of IDS and exist thereabout. Because the same Logical Atom exist in the same space (CNS, IDS, etc.) means contradiction.

4.2.2. Space created by representative atom

RPA set of CNU is called Consciousness Space (CNS), RPA set of PBU is called Probabilistic Space (PBS), RPA set of NMU is called Normalization Space (NMS), RPA set of RPU is called Representative Space (RPS), and the RPA set of NTU is called Natural Space (NTS). Other Spaces than CNS happen to be generally called CGS. Viewed from CGU, CNU created by only one theorem is always established. PBU, NMU, and RPU play a role of establishing correspondence between CNU and NTU. As CNU represents an idea of whole, its characteristics are denotative. As CGU represents an idea of part, its characteristics are connotative. Thus, CNS represents a denotative world and NTS represents a connotative world.

5. Intention

By determining a pair of NTU that starts Coupling, Coupling starts. Coupling repeats endlessly. RPA of NTU established by Coupling eventually becomes Boundary Atom. The NTU is called Critical Unit. On the other hand, Boundary Atom can become RPA of CNU. This CNU is called Singular Unit. In this case, both Critical Unit and Singular Unit take the same Boundary Atom as their RPA. It means that correspondence is established between these two Units, although they previously had no characteristic correspondence. The world established by Critical Unit contains contradiction whereas the world established by Singular Unit does not. In other words, the world with contradiction is replaced by the world without contradiction due to this correspondence. This is a state in which Intention is established. In this case, the number of CNA belonging to a subset to establish the Singular Unit is expressed by k.

As mentioned earlier, CNA belonging to a subset is called Attributes. In other words, the number of Attributes to establish the Singular Unit is expressed by k. A subset having CGA whose number is the same as k is selected from NTU among history of Coupling that has established the Critical Unit. Attributes that establish the selected

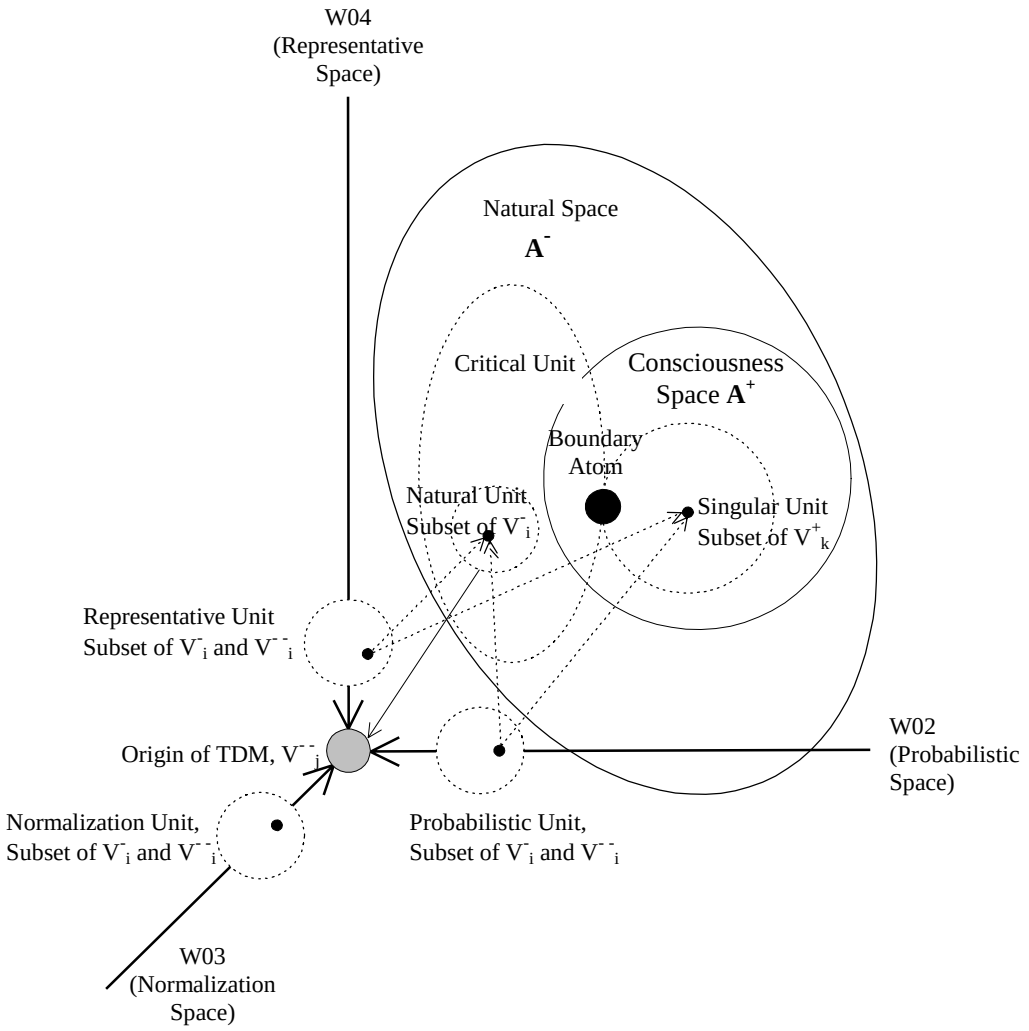


Figure 1. Three-dimension-like space model

NTU are objectified. We consider a result of this objectification as an object and the mode in our world is determined by objects. The objectification and the mode are established as a dynamic state. In our world, the objectification is a beginning of a dynamic state and the mode is determined by objectification.

6. Three-dimension-like space motel (TDM)

Three-Dimension-like Space Model (TDM) is a unified model of characteristics of CNS, NTS, PBS, NMS, and RPS. This is a model to give a base to establish "Intention," "objectification," and "the mode." In this study, Intention is positioned as a cause for objectification. We cannot cognize the Intention. What we can cognize is limited to a result of objectification or a state of the mode but not all of them. Objectification becomes a cause for the mode. For example, a result of objectification is Words and a state of the mode is all things in nature, our feelings, and our actions. These spaces (CNS, NTS, PBS, NMS, and RPS) have Logical Atoms as common elements. However, characteristics of Atoms in CNS and those in other four spaces are different as mentioned earlier. Unit is created from Atoms. A relation of five spaces unified in TDM is discussed by using Atoms and Units. RPA of the selected NTU becomes the coordinates origin of TDM. Attributes that establish this NTU can also become CGA of respective subsets that establishes PBU, NMU, and RPU (hereinafter referred to as 3 Units) according to theorems established in CSM. Because of this, it is considered that respective RPA of the 3 Units becomes the same as RPA of this NTU. Because of this, the coordinates system can be established. At the same time, this origin of the coordinates is the origin of three spaces (PBS, NMS, and RPS). In this case, NTS is positioned at a location created by this coordinates system. Meanwhile CNS is also is positioned at a location created by this coordinates system. This is based on theorems established in CSM. This is TDM and schema of TDM is described by Figure 1.

Hereupon, CNS positioned in TDM becomes a set of RPA that establishes Singular-CNU and NTS positioned in TDM becomes a set of RPA that establishes Critical-NTU. TDM defines a relation where Intention materializes, which means that correspondence between CNS and NTS is established by using the 3 Units (PBU, NMU, and RPU). A meaning of "using the 3 Units" is given by an operator deduced from theorems in CSM. A basic structure of this operator is called Predicate Structure. This Predicate Structure becomes a proposition and plays a role of "using." [2]

7. Predicate structure

Figure 2 indicates Predicate Structure. Predicate Structure is comprised of three decision boxes and four procedure boxes. Descriptions to satisfy rules defined by theorems in CSM are placed in Box 1 to Box 4. Descriptions to satisfy rules defined by architecture of a current computer are placed in Box 5 to Box 7. Description is done by programming languages. Rules defined in each box are referred to as follows: rule in 1st Box is 1st Rule, rule in 2nd Box is 2nd Rule, et al. Rules to satisfy Box 2, 4, 5, 6, and 7 require memory areas. Likewise, these areas are referred to as Area 2, 4, 5, 6, and 7 respectively. Predicate Structure creates two types of propositions. One is one-variable proposition by taking one word as Variable and the other is plural-variable proposition by taking multiple words as Variable. When a value is already set in 4th Area, this state is considered as the state of TRUE of the proposition. Details of Predicate Structure are discussed as follows. [3]

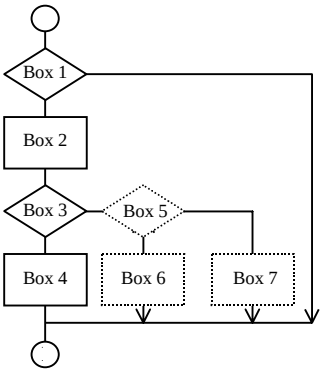


Figure 2. Predicate structure

7.1. Our providence

As mentioned earlier, there are five Units. Once PBU and CNU are determined, the remaining three Units are consequently determined. A relation between CNU and the other four is that CNU is always determined when viewed from the standpoint of the other four Units. In other words, PBU becomes the trigger to determine the remaining three Units. As a result, the three Units reflect PBU. And, CNU gives a rule to this reflection. Namely, four CGU are in a relation where they are created in the order from PBU, NMU, RPU, and NTU. Therefore, under this order, Occupied Spaces of RPA of all Units are in the following relation. The smallest is RPA of PBU and the largest is RPA of NTU. RPA of NMU and RPA of RPU are between these two in this order. The size of Occupied Space of RPA of CNU is positioned in between that of PBU and that of NMU. There are cases in which Occupied Space of RPA of RPU and that of NMU are equal. This is what we call Providence.

7.2. Significance of predicate structure

An operator to establish TDM is Predicate Structure. NTU having the same number as the number of Attributes

that establish the Singular-CNU is selected from plural numbers of NTU materialized by the Coupling. In this case, Attributes of the selected NTU exist among Attributes that establish the Singular-CNU. The above-mentioned relation is represented by Predicate Structure. Attributes of the selected CNU become Attributes of the PBU, NMU, and RPU. And Predicate Structure is created based on Attributes of the PBU, NMU, and RPU.

7.2.1. Predicate structure materializing in Probabilistic Space (PBS)

TDM is established by considering RPA of PBU, NMU, RPU, and NTU as the same. Instead, a relation of our Providence is given to Significant Space created by Attributes that establish each Unit (PBU, NMU, RPU). From this, Attributes that establish the PBU are selected in a manner where Significant Space of the PBU becomes the smallest among that of NMU, RPU, and NTU. These CGA should exist in NTS. The CGA existing in NTS are Attributes that establish Critical-NTU. The above-mentioned relation is represented by the 2nd rule of the Predicate Structure. The PBU can be established as long as it is assured that such CGA exit in NTS. The 3rd rule checks whether or not the CGA exit in NTS. If PBU is established, the 4th rule declares such fact. The 1st rule always plays a role of confirming this declaration. If it is not declared by the 4th rule, the 2nd rule defines PBU. If it is declared, there is no need for determining PBU again.

7.2.2. Predicate Structure materializing in Normalization Space (NMS)

In this study, we hypothesize CNS, CNU, CNA, CGS, CGU, and CGA. Among them, we can materialize only CGA. It is Words. NMU is established when the size of Occupied Space of RPA is larger than that of CNU. However, we cannot cognize the size of Occupied Space of CNU as mentioned earlier. Instead, it is compared with RPA of PBU in TDM and if it is larger than RPA of PBU and/or it is compared with RPA of RPU in TDM if it is smaller than or equal to RPA of RPU (discussed later), NMU is considered to be established.

The above-mentioned relation is determined by a subset that materializes the NMU. The Attributes should be Attributes that establish Critical-NTU. In this case, our Providence mentioned earlier should be satisfied. That is, Significant Space of a subset that establishes the NMU should be larger than Significant Space of a subset that establishes the PBU. And, it should be smaller than or equal to Significant Space that establishes RPU (discussed later). The above-mentioned relation is represented by the 2nd rule of the Predicate Structure. The 3rd rule checks if this comparison is established, which means if CGA exist in subsets that establish PBU and/or RPU. If NMU is established, the 4th rule declares such fact. The 1st rule always plays a role of confirming this declaration. If it is not declared by the 4th rule, the 2nd rule defines NMU. If it is declared, there is no need for determining NMU again.

7.2.3. Predicate Structure materializing in Representative Space (RPS)

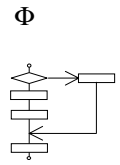
RPU is created based on NMU as its origin. As mentioned earlier, Coupling takes place among RPU. As a result, new RPU is created by this Coupling. In RPS, RPU created by NMU and RPU created by Coupling exist together. RPU used in TDM is RPU created by NMU. It is unnecessary to think about RPU created by Coupling. Because the objectification and the mode are positioned as life phenomenon in terms of cognition. Therefore, in this study, the objectification and the mode are eliminated from an issue of our modeling. TDM grasps a state until immediately before objectification occurs. Coupling is a cause to create the objectification and the mode. In TDM, RPU created by Coupling is eliminated. Attributes that establish RPU are selected freely from Attributes that establish NMU. However, our Providence mentioned earlier should be satisfied. That is, Significant Space of a subset that establishes RPU should be larger than Significant Space of a subset that establishes NMU. And, it should be smaller than Significant Space of a subset that establishes the selected NTU.

The above-mentioned relation is represented by the 2nd rule of the Predicate Structure. The 3rd rule checks whether or not the Attributes that establish the RPU are Attributes that establish NMU and/or CGA belonging to RPS. In CSM, the 3rd rule should have been “to check if it is Attributes that establish the selected NTU.” However, it is replaced by “to check if it is CGA belonging to RPS” as mentioned earlier, because we cannot cognize the former CGA. If RPU is established, the 4th rule declares such fact. The 1st rule always plays a role of confirming this declaration. If it is not declared by the 4th rule, the 2nd rule defines RPU. If it is declared, there is no need for determining RPU again.

8. Scenario function (SF)

TDM itself cannot be executed on a computer. In order to execute it on a computer, TDM is expressed in a program language. This is called SF. A significance of executing SF is not the same as a significance of executing conventional programs. However, a result of executing SF produces the same state as Intention of conventional programs. The mode of executing SF is not the same as the state of SF on a desk whereas the mode and the state become the same in conventional programs. Execution of SF is not conducted to materialize functionality purposes

like that of conventional programs. Execution of SF on a computer means to establish boundary between denotative and connotative worlds. Denotative world is a world expressed by CNS. Connotative world is a world expressed by NTS. Boundary is considered as a set of the coordinates origin of TDM. It means that these are distinctively different from the structure of conventional programs. Execution of SF establishes three spaces that are to become the coordinates, which means to determine the coordinates origin. Continuous execution of SF means to create a set of the coordinates origin. SF is a unique universal structure like TDM. Definition of SF is as follows.



SF= Φ [Φ4({(L4,j)},{(O4,r_α)},{(S4, r_γ)},R4)
+Φ2({(L2,i)},{(I2,r_β)},R2)
+Φ3({(L3,j)},{R3r, R3d, R3m, R3c})]

The meaning of this expression is represented by Figure 3. *Φ* is called Tense Control Function and *Φ4*, *Φ2* and *Φ3* are called Pallet Function. *L4,j*, *L2,i* and *L3,j* are generally called Signification Vector, whereas *O4,r_α*, *I2,r_β*, *S4,r_γ*, *R4*, *R2*, *R3r*, *R3d*, *R3m* and *R3c* are generally called Action Vector. Signification Vector and Action Vector are called Tense Control Vector in a generic term. Tense Control Function and Pallet Function are considered OS in conventional programs. That is, these are grasped as given Existence. Tense Control Vector is made of Predicate Structure. Signification Vector is determined by using one noun as a variable of Predicate Structure. The noun exists in Requirement. Signification Vector means one-variable proposition. Action Vector is determined by a set of nouns of Predicate Structure. Explanation of Figure 3 is as follows. Tense Control Function, Pallet Function, and Tense Control Vector comprise three layers. Execution of SF means that Execution Right is passed over from *Φ* to Pallet Function as a result of executing *Φ*. The Execution Right is passed over to *Φ4*, *Φ2*, and *Φ3* in the order. Execution in each Pallet Function is explained as follows. Execution of *Φ4* is to execute all *L4,j*, *O4,r_α*, all *S4,r_γ*, and *R4*. In this case, Execution Right is passed over to each Tense Control Vector from *Φ4*. Each Tense Control Vector returns Execution Right when execution is terminated.

Φ4 judges if the above-mentioned execution of *Φ4* itself should be repeated. If necessary, the above-mentioned execution is repeated. If not, *Φ4* returns Execution Right to *Φ*. As mentioned above, *Φ* passes Execution Right to Pallet Function that should be executed next. The next Pallet Functions to be executed after execution of *Φ3* is terminated are as follows.

- *Φ4* belonging to the same SF (*R3r*)
- *Φ4* of SF at lower level that is next to SF where the *Φ3* belongs (*R3c*)
- *Φ3* at higher level that is next to SF where the *Φ3* belongs (*R3d*)
- *Φ4* at higher level that is far away from SF where the *Φ3* belongs (*R3m*)

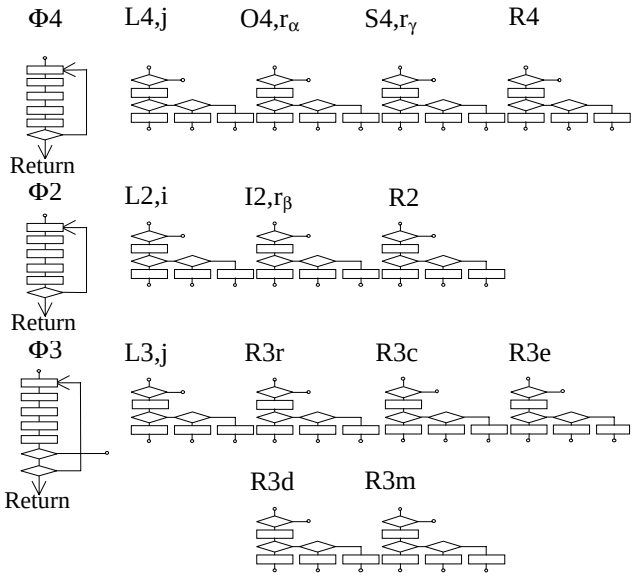


Figure 3. Scenario function

9. Requirement

In this study, we consider that we cannot question if Requirement is true or false. If we could, it should be fine too. However, we consider that Requirement is uncertain by nature and a role of software is to determine Requirement in a way it is convenient for human through a computer in cooperation with human. In order to do that, human, software, and computer should allocate roles/responsibilities. To date, this has been carried out. However, human in this case means mainly users of the system. We consider that it should also be convenient for human who develops software or who maintains the system. That is, software that is convenient for human who develops and who uses it should be pursued. We should grasp Requirement from this standpoint. Thus, it is necessary to pre-determine a final structure to determine program, and pursue Requirement.

Requirement cited herein is defined as a sentence in natural language except for programming languages (simply, sentence hereinafter). Mathematical expressions are considered to be included in natural language herein. Words existing in Requirement become Variables of

Predicate Structure. For example, parts of speech such as verb, adjective, preposition, and adverb are not Words herein. Roles of parts of speech other than Words are considered to be established by the complementary action materialized by executing SF. Words that cannot still be established regardless of the above-mentioned action are established by new concepts. The new concepts are expanded concepts of Words. There are three kinds of expanded concepts: Boundary Word, Representative Word, and Collective Word. Explanation of the concepts can be viewed at http://www.lyee.co.jp.

10. Synchronous structure

Since we hypothesize that the above-mentioned concept of Coupling creates time, Intention is defined as a correspondent relation between a world where time is established and a world where time is not established in this study. A relation to establish Intention and Requirement that reflects the Intention is called Synchronous Structure. TDM or SF represents a unit of Synchronous Structure. As Requirement materializes in NTS where time is established, the NTS where it materializes changes according to transition of time. There are cases where only a part of NTS changes and the whole changes. In case that a part changes, corresponding CNS does not change. The CNS is a world where time is not established. In case that the whole changes, corresponding CNS changes. Therefore, a relation in which the whole changes is called Asynchronous Structure.

TDM and SF should be re-defined separately if the whole changes. Requirement that we cognize is grasped by a plural numbers of SF. A relation of a plural numbers of SF is called Process Route Diagram (PRD) and is defined by a concept of only three kinds of routes. They are Continuous, Recursive, Duplex, and Multiplex. Partial change is grasped within the same SF. This relation is called Recursive, which is a concept of route. Details of a concept of route are not discussed herein. Case examples of PRD can be viewed at http://www.lyee.co.jp.

11. Example of scenario function

Apple is a concept established by a substance and Attributes. We cannot cognize a substance whereas we can cognize Attributes. Apple is considered a substance and red color, rounded shape, sweet and sour, et al are considered Attributes herein. One Singular-CNU is expressed as follows.

Apple = {Apple, red color, rounded shape, sweet and sour……..}

Meanwhile, selected CGU is expressed as follows.

Apple = {red color, rounded shape, sweet and sour……..}

In case of CGU, Apple does not belong to elements of the set. Apple becomes the origin of TDM. Elements of the set of CGU become elements of PBS, NMS, RPS, and NTS. An element of CNS is Apple.

For example, a relation of Apple in CNS, red color in NTS, and red color in PBS is proposition established in PBS herein. Likewise, a relation of Apple in CNS, red color in NTS, and red color in NMS is proposition established in NMS. This proposition is an action to belong to NMS. Likewise, a relation of Apple in CNS, red color in NTS, and red color in RPS is proposition established in RPS. This proposition is an action to belong to RPS. This proposition is defined for all Attributes. This proposition is determined by giving one Attribute belonging to each space to Predicate Structure. When all of the propositions of PBS, NMS, and RPS become True, it is considered that correspondence between CNS and NTS has been established. A state of True of each proposition is stored in a computer memory. Execution of SF autonomously determines whether or not it is True among propositions. SF has a relation like this. How to define the proposition is already discussed in this paper. Further details of actual case examples are viewed at our homepage.

12. Conclusion

The ultimate aim of an observation in this study is to prove that software can be determined by a universal deterministic manner. A deterministic manner will ultimately be determined by about eighty to one hundred theorems. Discussed herein is the proceeding phase to determine these theorems. Various kinds of software have already been developed by the algorithm determined in the proceeding phase. These can be viewed at our homepage. It is confirmed that if Requirement is the same, SF will be the same whoever makes it. Moreover, it is possible to automatically convert conventional programs (software developed by other methods than our algorithm) into SF. It is obvious that there is no universal algorithm to determine these conventional programs. Regardless of this, the fact that the algorithm to automatically convert conventional programs into SF is established indicates extreme significance. In summary, SF has already been determined by a universal algorithm. It indicates that a universal algorithm can be replaced by theorems. We are hoping to proceed with this process further.

Developing software by this algorithm may change understanding of conventional software world. For example, development procedure proposed by DOA will be dramatically simplified such as workload of designing phase and testing phase. Logic to establish software will also be dramatically simplified. Because things that have been determined by Modeling and Designing can be

deduced by execution of SF, which means the following. Human workload of software development can also be simplified by a universal algorithm. This thinking has been influenced during its course of discovery by the ideas of Spinoza, Leibniz, Wittgenstein et al, and in that sense, the software of this study is a new definition of software.

References

[1] F. Negoro, "Principle of Lyee Software", Proceedings of 2000 International Conference on Information Society in the 21st Century (IS2000), Aizu, Japan, November 5-8, 2000, pp.441-446.

[2] F. Negoro, "Intent Operationalisation for Source Code Generation", Proceedings of the 5th World Multi-Conference on Systemics, Cybernetics and Informtatics (SCI2001), Orlando, USA, July 22-25, 2001.Software Engineering Artificial Intelligence, Networking & Parallel/Distributed Computing (SNPD'01), Nagoya, Japan, August 20-22, 2001.

Proceedings of International Conference on Advances in Infrastructure for Electronic Business, Science and Education on the Internet (SSGRR2001). (L'Aquila, Italy, August 2001) .