

English Translation of the Published Application

International Application No.: PCT/JP97/01492

International Filing Date: 30 April 1997

International Publication No.: WO98/19232

International Publication Date: 07 May 1998

Please note that the translation of the Abstract attached hereto is not the published English Abstract on the front page of the published PCT application document. The attached Abstract was translated word for word from the Japanese Abstract as filed on 30 April 1997.

DESCRIPTION

SOFTWARE PRODUCTION METHOD, PROCESSOR, PROCESSING METHOD AND
RECORDING MEDIUM

Technical Field

This invention is related to the production method, processor, processing method and recording medium, which are applicable to all business software, games software, and software for all other fields.

Background Art

How well the OS or package may be made, they cannot fulfill user's needs by themselves alone. This is because software always contains the part which users have to develop on their own. The important point is that the software production technology must be adopted to the part stated above. Consequently, the production technology must enable to realize such effects as short-term software development, release from maintenance problems, methods of connecting the upper-stream specifications and the program specifications functionally or theoretically, and guaranteeing the program's functional quality.

The traditional method is a functional division method, by which requirement functions are extracted to be made into software. In most cases, however, it is not understood that a supplemental logic will eventually arise from the functional division method. Besides, it cannot go too far by

saying that this way of thinking has a fundamental flaw, which causes all software problems to happen. To realize the fundamental improvement of the problems, it is necessary to introduce a new concept or a theory to uni-vocalize the requirement event, thereby replacing the traditional way of thinking.

The inventor has contrived such a theory and named it Lyee (GOVERNAMENTAL METHODOLOGY for SOFTWARE PROVIDENCE). This theory provides the standard of a new handling method for semantics.

The inefficiency of software production maintenance and the instability of the development period directs software production means to a mechanical function packaging method that ignores the specific characteristics of software.

This package itself is also a non-correct solution type in that the package lacks the uni-vocalization. As a result, user's environment is gradually taking a worsening course, as an aspect of reliability, usability, and adaptability (RAS) functions. What is needed at this exact time is to prepare the base for the production.

In this field, a solution to problems of the production technology has not been tried, i.e., without having the method of managing the quality of the program-unit process function, yet it has been insisted that the good software can be decided by its upper-stream only. This is just the reason of abnormality. The software production method is not inherently of characteristic allowing the separation of the upper and lower stream concerns. We must start, however, at the point of grasping it theoretically. The Lyee theory exactly satisfies it.

That is, the purpose of the invention is as follows:

- (1) To improve the production ability of software developers and maintenance personnel.
- (2) To theoretically connect the upper-stream and lower-stream information.
- (3) To provide a program structure applicable to the development for the broad software field including not only business software but also OS, middle software, games software, etc.
- (4) To minimize the upper-stream information and to create a source program.

Disclosure of Invention

Title: A real state of the software structure

Fumio Negoro

The Institute of Computer Based Software Methodology and
Technology

Chapter 3 Development Methodology

This invention is what replaces Lyee with the activity method of the software development. Therefore, this invention directly reflects Lyee. In order to simply utilize this invention's effect, the implementation of works accorded in this invention's algorithm will do. Besides, the work method is simple, so it is not necessary to understand Lyee.

This does not mean that, in order to gain the effect of this invention, the understanding of Lyee is an absolute condition, if you recall the fact that the modern scientific levels have been maintained, although most scientific principles were not fully understood until now.

However, it is necessary to understand Lyee if one wishes to get full grasp of why the obtainable effect can be brought about. The essential points of Lyee will be mentioned herein as a help understanding the points above. An official treatise of Lyee would be required to be obtained separately.

Lyee is the world-first one that theoretically guarantees the principle of software formation.

In this field, up until now, the software has been produced by the way similar to the way of producing a structure. The present software problem as represented by the productivity, however, and the present countermeasure as represented by the packaging which underlays the problem stated above originate from the lack of recognizing of the event for making software.

It is necessary to understand that the software and the structure are different in principle in their way of existence.

An example is explained briefly hereunder.

The requirement event exists intrinsically, and the way of its existence is semantic.

In case of the structure, its requirement event can be expressed as a drawing and it can be made into an object by technical means.

Both existing ways of a drawing and of an object are extrinsic; the drawing exists as a semantic event and the object as an structural event.

Therefore, to bring out reversibly a drawing from an object can be realized at a high performance rate, if not perfectly. This guarantees the theoretical basis of grasping an

engineering subject as a relation between the drawing and the object and finding an overcoming means thereabouts.

On the other hand, in case of software, the one equivalent to a drawing is a design document, and the one equivalent to an object is a program. What is different from the structure's case is that an object will also become the same semantic event. This can be easily understood an easy understanding that technical means are difficult to be established thereabouts, if you recall that it is impossible to find a structural relationship between one meaning and the other meaning.

Also, if it is understood that making a design document from a program is difficult, it can be easily presumed that the software problem is a proposition that cannot be grasped only as a relationship between a design document and a program.

From what is set forth above, it is very clear that the software problem is not of characteristic of an engineering proposition's being able to be overcome by the empirical rule as of the structure. In other words, as a natural consequence, the software problem will eventually be a proposition that can only be solved by theoretical means.

Lyee considers intrinsic requirement events and extrinsic existence events as events of the same quality.

That is, in the case of taking us as a standard, existence events are meant to be what we are conscious of, which includes universal events, quantum-theory events, biological events, on-earth natural events in the real space where we exist.

In this connection, in Lyee, existence events such as meanings, sounds, fragrances, personalities are intrinsic events, and the sun, drawings, and programs are extrinsic events.

On the other hand, if we are not taken as a standard, for example, as an existence event which we will be able to be conscious of in the future is already an existence event, additional existence events should be in existence. Lyee treats all such events as an existence event.

Lyee is a theory originally established by Fumio Negoro, the inventor of this invention, and it is the theory for elucidating the only one cause of the existence event.

If the existence event is rephrased as the real aspect, the only one cause is the true state of things.

This work method gives a means for modifying the true state of things into specifications and for making them into software.

The requirement event in the real space will bring forth the diversity. If a space representing its true state is established, however, the true state can be obtained as the only one cause.

In Lyee, the space is called a consciousness space as compared to the real space, and a relation connecting the real space and the consciousness space is called the providence of existence.

Different from the real space, there exists no dimensional concept in the consciousness space.

Details are discussed in the Lyee treatise. In brief, Lyee is a theory to grasp the relation.

The formal name of this theory was given as GOVERNAMENTAL METHODOLOGY for SOFTWARE PROVIDENCE. Lyee is an abbreviation.

Lyee designates an operator coexisting in the two spaces of the real space and consciousness space as a significance and quantizes it. Although a means for making it into a quantum significance is not referred to herein, it is called an elementary, which is explained in detail in the Lyee treatise. Theoretically, this elementary is a minute particle having the maximum space with the diameter of 10^5 minus 100,000 digits.

The elementary builds an additional significance by congregating each other. The congregated elementaries are

called a chain. The elementary exists eternally, with no longevity, whereas the chain has its lifetime. The lifetime is one of the chain's significances. Details are explained in the Lyee treatise.

The elementary, accompanied by its rule (called an elementary rule) and two values (called a significant element), emerges into the idea space for taking a seat. This is the beginning of the formation of the existence event. The two accompanied significant elements are called a birth sequence and a time speed, both given by the natural number.

However, these values themselves differ from the values ruled by the dimension of the real space. Details are explained in the Lyee treatise.

The elementary rule specifies a space (called a significant space) and the other significant element which rules the elementary, at the same time co-lives in the significant space together with the two accompanied significant elements and the other specified significant element. The elementary exists through such process.

The significance is the minimum element composing an existence event.

The significance can be grasped as a meaning in a broad sense.

The theoretical elementary (or aspect) cannot be realized as

a concrete form. Therefore, in the work method advocated herein, the theoretical elementary is replaced by the words existing in the requirement event.

In this connection, terminal information, control telegraphic message, operational button, etc. are also used as a word.

The existence event is what the consciousness space's chain is associated and realized into the real space in conformity of the existence providence. In this connection, we are one of the existence events.

In the Lyee treatise, the existence event is defined as a group of unitized event chains. The embodying mechanism is set forth in detail in the Lyee treatise.

That is, Lyee correlates the true state and the existence event as having the elementary-chain structure.

Fig. 46 shows a structure model of the idea space created by the elementary which emerges into the idea space. This structure is explained in detail together with Fig. 46 in the Lyee treatise.

In the idea space, the elementary, after being made into a congregation, becomes a chain, thereby enhancing the significance.

The chain is classified into a definite chain, a consciousness chain, and a nature chain.

The build-up mechanism of this chain is explained in detail in the Lyee treatise.

These chains establishes one stream based on the existence providence, and subsequently creates an event chain, which will become an element of the existence event. And, they will develop into a group and be unitized.

The unitization is meant to be to become an existence event after being realized in the real space.

Fig. 47 shows a model of the above-mentioned relation.

This relation is called a rule structure.

The rule structure is explained in detail together with Fig. 47 in the Lyee treatise.

In this connection, the scenario function, which will be discussed hereafter, represents a structure of the consciousness space grasped from the point of the real space.

Lyee expresses this rule structure by a state expression.

This is called an associative equation.

The associative equation is not referred to herein, but it is set forth in detail in the Lyee treatise.

The associative equation gives a scheme by which the existence event is constructed based on the existence providence. That is, it grasps a formation relationship between the true state and the existence event.

Therefore, when the associative equation is reversibly solved,

it is possible to obtain the only one structure which the real space's existence event takes in the consciousness

space.

That is, it is possible to know what kind of chain the unitized chain group, representing the existence event, will be in the consciousness space.

In Lyee, the existence event was placed on the computer software, and a reverse solution was implemented.

The reverse solution is not referred to herein, but it is explained in detail in the Lyee treatise.

Fig. 48 shows a space structure used for the reverse solution. This is called a space paradigm. For the reverse solution of the associative equation, the space paradigm like this becomes indispensable.

To mention repeatedly, the space paradigm shown in Fig. 48 is applies only to the case of the event of making software. When the reverse solution is implemented with regard to another event, another space paradigm will be needed.

In the case of the event of making software, the real space will be expressed as three kinds of a two-dimensional surface, and the consciousness space will be expressed as a space created after the two-dimensional surface being synthesized. Then, in the consciousness space there exists an elementary having the consciousness space's significance, and in the cognitive space there exists an elementary having the cognitive space's significance. As the elementary will make chained,

and chains created thereabouts will become constructed by the elementary existing in the respective spaces. And, as aforementioned,

with this work method, the elementary can be replaced by words.

The relation of the word, its chain, and the existence providence, which all exist on this space's paradigm, is expressed as an associative equation, and its reverse solution is to be implemented.

The resulting expression is called a scenario function, which is shown as below.

$$T\ 0 = \Phi 0\ (\{ \Phi P\ \{ Li, 2\ | \ \{ T1, f \} \} k \} + \Phi P\ \{ Li, 3\ | \ \{ T1, g \} \} + \{ \Phi P\ \{ Li, 4\ | \ \{ T1, q \} \} k \})$$

The above expression indicates the state which is represented by the software existence event in the consciousness space.

The traditional program takes, from the beginning, an execution-format logical structure.

In other words,

the traditional software development ends up making a logical structure on which the computers should faithfully operate.

The scenario function, however, takes a static structure.

And, when this scenario function is made into a program and

executed on the computer, the result to be obtained is the same as when the traditional program has been implemented. The structure of the scenario function is fundamentally different from that of traditional program.

The scenario function represents the one and only paradigm ruling the software event of the computer. In other words, any computer software can be realized by using this function. This, too, is the world-first one.

The Scenario function's expression method $\{ \alpha \}$ indicates a congregation body of the grouping element α .

Explanation of the items of the Scenario Function

01. Definition of screen main W02 pallet having identifier

$k: \Phi P\{Li, 2 \mid T1, f, 2\}k$

02. Definition of main W02 pallet having a screen identifier

$k: \Phi P\{Li, 4 \mid T1, q, 3\}k$

03. Definition of the main W03 pallet (only one):

$\Phi P\{Li, 3 \mid T1, g, 4\}$

04. T0 is a scenario function ruling the main pallets W02, W03, and W04

05. Scenario function ruling the subordinate pallet of the object file f , which occurs in the main W02 pallet:

$$T1, f, 2 = \Phi 1(\Phi P\{Li, 2\}f + \Phi P\{Li, 3\}f + \Phi P\{Li, 4\}f)$$

06. Scenario function ruling the subordinate pallet of the object file g , which occurs in the main W03 pallet:

$$T1, g, 3 = \Phi 1(\Phi P\{Li, 2\}g + \Phi P\{Li, 3\}g + \Phi P\{Li, 4\}g)$$

07. Scenario function ruling the subordinate pallet of the object file q , which occurs in the main W04 pallet:

$$T1, q, 4 = \Phi 1(\Phi P\{Li, 2\}q + \Phi P\{Li, 3\}q + \Phi P\{Li, 4\}q)$$

The three kinds of the two-dimensional surface expressing the real space were discussed previously, and the pallets with the two-dimensional surface are expressed by the symbols W02, W03, and W03, respectively.

On the pallet, a base logic and a scenario function of the object file are installed.

The main and subordinate pallets take the same structure, but there are differences as follows.

The number of words of main pallets W02, W04 having screen identifier k is the same as the number of words existing in the screen identifier k . The number of words of the main W03 pallet is the same as the number of words satisfying all the events. On the other hand, the number of the words of the subordinate pallets for W02, W03 and W04 are all the same, which equals to the number of all the words satisfying the object files.

T0 is a program to process the on-line software event, and T1 is a program to rule the off-line software event. T1 can be assembled into T0 as shown by T0, whereas it can also be independent.

To mention a relative relation between Lyee and the traditional program, the pallet can be understood as resembling the program.

However, the scenario function is a concept to grasp the software event, so it is a paradigm which can grasp one subsystem, one function, or one traditional program, or all existing software events, by the only one function.

08. Φ_0 is a program to control the execution order of the main pallet, and it is called a pallet chain function. Fig. 19 indicates its logical structure.

09. L_i, j is a program to be decided by the word identifier (i) and the pallet identifier (j), and it is called a tense control vector. Its structure is shown in Fig. 12, 13, 14, 16, and 17.

10. Φ_p is a program to group the tense control vector with j and k, and it is called a pallet function.

Its structure is shown in Fig. 18.

11. Φ_1 is a program to control the execution order of the subordinate pallet, and it is called a pallet chain function. Its structure is shown in Fig. 15.

The structural specifications of the program composing the scenario function is an accumulation of solution conditions emerging during the process of inducing the scenario function. This is nothing but the grounds endorsing an advocacy that the software can be created by theory.

To proceed with practical works, a homogeneity map is produced.

The homogeneity map is a diagram abbreviating a scenario function by the way stated below and the map is to grasp the

relation between the pallet chain function and the definitives (screen, voucher, secondary memory file).

That is, $T0 = \Phi0(\{\{f, 2\}\}k + \{g, 3\} + \{\{q, 4\}\}k)$

$\{\{f, 2\}\}k$, $\{g, 3\}$, $\{\{q, 4\}\}k$ are tense control vectors, expressing a pallet not inclusive of T1.

That is, in Lyee, the screen, voucher and file are called definitives, and it expresses the relationship between their identifiers and the pallet chain function. The information amount exhibited by the homogeneity map is equivalent to 70% of the information amount included in the whole of what are called requirement definition, basic design, detailed design, program specifications, test specifications and operational manuals in the traditional method. If this advocated work method is utilized with this concern, the documentation work is decisively reduced, cutting down to approximately a one-twentieth.

What is called the tense control vector in this theory can be created from the homogeneity map and the function of words belonging to the definitive. As already referred to, in case of Lyee, six or five pieces of the tense control vector can be created from one word, and this corresponding relation is the one and only method of realizing a complete independence of the tense control vector. This context is explained in the Lyee treatise in detail.

There will be no other means to actualize a complete software componentization. The traditional way of componentization has been simply a functional partialization, and as a consequence, a strict sequence must humanly be attached to its logical chain.

The effect of the componentization cannot be obtained from such structure.

The tense control vector is a general nomenclature of the duplicate vector and the homogeneity vector.

And, the user's logic can be fulfilled only by building up this tense control vector. As already mentioned, the documented material amounting to 1,000,000 lines, 100,000 pages, which are common in the traditional method can be decisively eliminated.

The tense control vector is identified by a classification between the homogeneity vector and the duplicate vector, a word identifier, an identifier of definitive to which it

belongs and a pallet identifier, and it is grouped by the pallet function based on the identifiers.

The pallet is a body of a group of the tense control vector, and at the same time, it is a body of a group of the data area whose address is the identifier.

The tense control vector creates only a self data by its logic, and it is a program to actualize the function of setting the self data into the self's data address.

In case of the homogeneity vector, by using only data in the data area of a pallet to which it belongs, it creates self's data by its logic and sets it into self's address.

In case of the duplicate vector, it phase-shifts data from the data area of the nearest pallet shown by the homogeneity map to self's address. The address of the obtained data area is limited to the address of the same word identifier. The nearest pallet is crystalized as the one and only.

If no data exists in the nearest pallet, the data in the next-nearest pallet's data address is phase-shifted.

If no data exists in either pallet, the duplicate vector is not justified.

In this connection, in case of a system with the scale of 1,000,000 lines, the approximate number of words appearing thereabouts is 2,000. In this instance, the number of the tense control vector built up by users will be 10,000 pieces. The average number of logical lines is to at least and 20 at most per tense control vector: the average number of logical lines for the case above will be 100,000 at least and 200,000 at most.

As the logic of the tense control vector is simplified, the productivity can remarkably be enhanced, resulting in 10,000 to 50,000 lines/man-month. The total of works is ten to two man-months.

The data set established in the tense control vector always comes to a state to be homogenized. This is a feature that is performed by the tense control vector.

The concept of data homogeneity is a concept which one should mention as a principle of simplifying software events to the greatest extent. It is what was won by Lyee. That is, when we observe what we call a function from the point of data homogeneity, the function is deducted to be a chain of non-homogeneity data. And, if one recollects that the creating a logical function from the requirement event leads to dividing the meanings and that the division of meanings is an impossible task, it can be easily imagined that the working method so far which has been realizing the functionalization by using this way of dividing meanings equals to what causes the thermal death as explained by the second law of thermodynamics.

Although figurative, it is the most suitable to entrust users with the event of making software,

not with experienced SEs. This is because the sincerer SEs with abundant experience are more likely to create the thermal death. The situation that the productivity of this filed shows a year-wise drop is attributed to this reason.

The only means to avoid it is a data homogenizing concept. This concept is explained in detail in the Lyee treatise. In other words, the scenario function is to regulate the logical structure building up a group of data to be homogenized. And, repeatedly, it can be stated that when it is executed on the computer, a chain of non-homogeneity data, that is, a function, will be created.

Therefore, only the homogeneity data will do for the requirement event to establish the scenario function.

As understood well by the after-mentioned example, the reason why this working method defines the requirement event is fully satisfied only by the definitive and its words is that the scenario function takes the structure which is fulfilled only by those.

Theoretically, no logic exists between the homogenized data group each other, but only a memory activity with no logical sense will be generated. That is, no function other than a memory activity cannot exist thereabouts.

This is explained in detail in the Lyee treatise.

It is from this reason that the volume of the program to be developed by this work method can be reduced to one-fifth those developed by the traditional method.

All mentioned above is an outline of the relationship between the Lyee theory and the present invention. The Lyee theory does not cause the gap between actual development work and the theory. It has its strict influence upon even one line of the program comprising the scenario function, and accomplishes creating a logical structure unprecedentedly good not only as the source program standard but as the execution type.

-COMPLETE-

Fig. 47 Supplemental Explanation

01. The grouped consciousness elementaries make a row in the ascending order of its birth sequence.
02. The grouped cognition elementaries make a row in the descending order of its significant space size.
03. The chain internally contains the time providence (growth time, grouping time, chain time, transition time, life time, transposition time, recurrence time).
04. The interval value between birth sequences whose elementarie's are neighboring each other is called a deuteronomous time of the latter elementary.
05. The chain built up as a complementary relationship expresses a transition of chain.
06. The consciousness chain is all made anew every time the distribution structure is rebuilt.
07. As for the cognition chain, only the cognition chain to be affected is dissolved when the convergence structure is rebuilt.
08. The significance which bases elementary are a birth sequence and a time speed.
09. The chain is composed of an elementary's partial set, the elementary's permutation, and an equivalence elementary.
10. The equivalence elementary expresses a significant space of the chain.
11. A set of the consciousness chain's significance space is called a consciousness space.

12. The cognition chain is a generalized term for a establishing chain, a nature chain and an event chain.
13. A set of the definite chain's significant space is called a definite space.
14. A set of the nature chain's significant space is called a nature space.
15. A set of the event chain's significant space is called an event space.
16. The elementary rule activated by the distribution rule builds up a consciousness elementary's partial set.
17. The elementary rule activated by the convergence rule builds up a cognition elementary's partial set.
18. A rule created by the consciousness elementary's partial set is called a consciousness rule.
19. A rule created by the cognition elementary's partial set is called a cognition rule.
20. The consciousness rule activated by the distribution rule builds up a consciousness chain.
21. The definite rule is activated with the contingency by a definite rule in a broad sense and builds up a definite chain.
22. The nature rule is activated by the consciousness chain and builds up a nature chain.
23. The nature chain is a broadly-meant nature rule and builds up an event chain.
24. The event chain is a breakup of the nature chain.
25. A plurality of event chains constructed by one nature chain being split have been homogenized with the nature

chain.

26. The homogeneity of the consciousness chain constructed of two nature chains are inconstant.

27. An event chain activated by a broadly-meant event rule makes event chains grouped.

28. The rule created by the grouped event chains will unitize them by the contingency.

29. The unitization means that a set of event chains are materialized into phenomenon.

30. The existence providence means a stream as follows:
definite chain $\rightarrow$ consciousness chain $\rightarrow$ nature chain.

31. When the relationship between the recurrence = the equivalence elementary's diameter (R_K) of the definite chain and the equivalence elementary's diameter (R_W) of the consciousness chain is expressed as $R_K \geq R_W$,

the relationship between R_W and the equivalence elementary's diameter (R_C) of the nature chain built by the consciousness chain's associating is expressed as $R_C \geq R_W$ and,

if, in this instance, there exists an elementary α in the convergence structure, wherein α satisfies unequation $R_C > R_\alpha > R_W$,

then the elementary α can be replaced by an elementary of the distribution structure by gaining a new time speed.

32. The transposition means that a middle-point elementary is replaced by a cognition elementary due to a new consciousness elementary's appearance or recurrence.

33. The reverse transposition means a reverse action of

transposition.

34. The associative equation is what expresses existence providence by logical structure.

Brief Description of drawings.

Fig. 1 indicates the work process according to the present invention.

Fig. 2 indicates a screen on which to enter sales data.

Fig. 3 indicates a screen on which to refer the customer code.

Fig. 4 indicates a screen on which to enter the confirmation of arrival of goods.

Fig. 5 indicates a screen on which to refer to the warehouse code.

Fig. 6 indicates a screen on which to instruct the output of the delivery request list.

Fig. 7 indicates a voucher of the delivery request list.

Fig. 8 indicates a voucher of the delivery request list.

Fig. 9 indicates a homogeneity map for the "sales entry".

Fig. 10 indicates a homogeneity map for the "arrival confirmation entry".

Fig. 11 indicates a homogeneity map for the "delivery list".

Fig. 12 indicates one example of the W04 duplicate vector.

Fig. 13 indicates one example of the W04 homogeneity vector.

Fig. 14 indicates one example of the W02 homogeneity vector.

Fig. 15 indicates a structural paradigm of the pallet chain function (for off-line.)

Fig. 16 indicates one example of the W03 duplicate vector.

Fig. 17 indicates one example of the W03 homogeneity vector.

Fig. 18 indicates a structural paradigm of the pallet function.

Fig. 19 indicates a structural paradigm of the pallet chain function (for on-line).

Fig. 20 is a drawing to explain the scenario chain.

Fig. 21 indicates a whole structure of the software concerned with the present invention.

Fig. 22 indicates a structure inside the pallet.

Fig. 23 indicates a logic paradigm of the tense control

vector.

Fig. 24 indicates the characteristics of a logic by Lyee.

Fig. 25 indicates a structure of the traditional-type program.

Fig. 26 indicates a structure of the traditional-type program.

Fig. 27 indicates a module composition of the traditional-type program.

Fig. 28 indicates a structure of the Lyee-type program.

Fig. 29 indicates the W03 homogeneity vector.

Fig. 30 is a drawing to explain that Lyee handles only homogeneous data.

Fig. 31 indicates one example of the Lyee's W03 pallet.

Fig. 32 is a drawing to explain that the auto logic and the operational logic are separated in the Lyee software.

Fig. 33 is a drawing to explain the difference in the error correction procedure between Lyee and the traditional-type.

Fig. 34 is a drawing to explain the difference in the software development method between Lyee and the traditional-type.

Fig. 35 is a drawing to explain more specifically the development work of Lyee-applied software.

Fig. 36 indicates one example of the screen defining specifications.

Fig. 37 indicates one example of the voucher defining specifications.

Fig. 38 indicates one example of the file defining specifications.

Fig. 39 indicates one example of the homogeneity map.

Fig. 40 indicates the method of making the homogeneity map.

Fig. 41 indicates the homogeneity vector.

Fig. 42 indicates one example of the tense control vector specifications.

Fig. 43 indicates one example of the tense control vector specifications.

Fig. 44 indicates one example of programming of the tense control vector specification.

Fig. 45 is a drawing to explain the effects of the present invention.

The Best Mode for Carrying Out the invention

That is, in this invention, it is the first thing to decide a definitive identifier for the screen needed in the software to be produced. The second thing is to sort out words existing in the screen as well as to make a homogeneity map based on the definitive identifier, in which map all pallets necessary for the software are placed in

accordance with the process flow. The third thing is to decide a necessary file based on the sorted-out words and the homogeneity map made. The fourth thing is to make the first tense control vector to do screen edit or file edit, the second tense control vector to decide the process route, and the third tense control vector to update files, for all sorted-out data. The fifth things are to make the three kinds of pallets, each of which bundles by the first, the second and the third tense control vector, respectively, as well as to make the pallet function, which executes each tense control vector with significance self-ruled in each pallet. The sixth things are to transmit the screen based on the pallet function concerned with the first tense control vector, to receive the screen to execute the pallet function concerned with the second tense control vector, and to assemble, in accordance with this execution's result, the aforementioned three pallet functions into the pallet chain function, which has a structure to decide one process route among a plurality of process routes taking the execution of the pallet function concerned with the third tense control vector as at least one process route.

In the following, in order to explain the present invention in detail, the explanation is presented in accordance with the attached diagrams.

Fig. 1 indicates the work process implemented by the present invention.

That is, in the present invention, a desired software can be created by the following steps: to decide the definitive identifier (step 101); to decide words (step 102) as well as to create the homogeneity map (step 103); to collect the tense control vector (step 104); to decide the file (step 105); to make the tense control vector (step 106); to make pallet function (step 107); and to assemble them into the pallet chain function (step 108).

Determination of the definitive identifier

For example, in the case of creating software called "sales entry," which is for inputting sales data in a certain corporation, the screen (Fig. 2) on which to input sales data and the screen (Fig. 3) on which to refer the customer codes are to be decided. Further, with the tense control vector, there are five (or six, in some case) tense control vectors corresponding to each of these words. Not only the customer code reference screen but also such screens can also be set if needed as a staff code reference screen, a dealing classification screen, a billing-to codes screen, a product codes screen, etc., similar to the screen stated above.

Also, in the case of creating software called "arrival

confirmation" which is for inputting the confirmation of product arrivals in the other company, the screen (Fig. 4) on which to input the confirmation of the product arrivals and the screen (Fig. 5) on which to refer the warehouse codes are to be decided.

Also, in the case of creating software called a "delivery request list" which is for outputting a list of product delivery requests in the other company, the screen (Fig. 6) in which to control the output of the delivery request list and the list's vouchers (Fig. 7, Fig. 8) are to be decided.

Determination of the word

The determination of the word means to sort out all the words from the above-decided screens and vouchers.

For example, in Fig. 2 screen which is for the "sales entry," the following words can be sorted out:

OPCD

Sales No.

Sales Classification

Sales Date

Customer

Payment Due Date

Billing Destination

Consumption Tax

Staff in Charge

Slip-base Order No.

Delivery Location

Phone Number

Remarks

Product Codes

Quantity

Unit Price

Discount

Amount

Product Name

Type of Machines/Media

Article Number

Sales Total

Consumption Tax Total

Discount Total

Sum Total

Execution

F3

F4

F8,

and these are the determined words.

Also, in the above "customer code reference" screen of Fig. 3, although not illustrated, there are such words as follows:

No.

Customer

Customer (abbreviation)

Customer Name

Address

Staff in Charge

Selection No.

Customer Name in katakana

Old Code

Execution

F12

The determined words are what were sorted out from these words.

Also, in the above "arrival confirmation entry" screen of Fig. 4, although not illustrated, there are such words as follows:

OPCD

Warehouse Codes

H/S Classification

Usher Classification

Execution

F1

F3

F4

The determined words are what were sorted out from these words.

Also, in the above "arrival confirmation entry" screen of Fig. 5, the following words can be sorted out:

No.

Warehouse Codes

Warehouse Name

Selection No.

Execution

F12

and these are the determined words.

Also, in the above "delivery request list" screen of Fig. 6, the following can be sorted out:

Delivery Request Date, Execution

F2

F3

and these are the determined words.

Also, with the vouchers of the above "delivery request list" of Fig. 7 and Fig. 8, the following can be sorted out:

Department

Delivery Destination

Billing Destination

Address 1

Address 2

Phone Number

Dept. 1

Staff in Charge

Staff in Charge of Store

Delivery Request Date

Delivery No.

Product Code

Product Name

Quantity

Update

Order Receive Date

Order Receive No.

Line

Staff in Charge of Order Receiving

Article Number

Date

Your Company's Order No.

Inst.

City

Delivery

Sales

These are the determined words.

Making of the homogeneity map

The homogeneity map is made based on the determined definitive (screens or vouchers) identifier. The homogeneity map may be made by a worker or be automated mechanically. On the homogeneity map, pallets (indicated with a box) corresponding to each screen and each voucher are connected each other with lines, and the connection accords with the regulation of the pallet chain function. The line means the pallet chain function. If screens and vouchers have already been determined, the connections between pallets are to be necessarily determined.

The Fig. 9 is a homogeneity map for the above "sales entry."

91 is a menu screen. One can select "sales entry" from

the menu screen.

92 is the W04 pallet for the screen (Fig. 2) on which to enter sales data, whereas 93 is its W02 pallet.

94 is the W03 pallet at the time when "execution" has been selected in the screen on which to enter sales data.

95 is the W03 pallet at the time when "registration" has been selected in the screen on which to enter

sales data. The WFL (Work File Area) 96 means to write data onto a file.

97 is the W04 pallet for the screen (Fig. 3) at the time when "reference" of "the customer code" has been selected in the screen on which to enter sales data, whereas 98 is its W02 pallet.

When "end" is selected in the screen on which to enter sales data, it returns to the menu screen.

In each pallet, the following are described: the kind of pallet (in the upper part outside the box); the name of screen (in the upper part inside the box); the screen identifier (in the lower part inside the box); and the determined file name (discussed later) (in the dot-line box in the lower part outside the box).

Above the line connecting pallets, the name of buttons (such as "execution" and "registration") are described, and below the line is described, the absolute classification of the process (R0 - R5) (discussed later).

Fig. 10 is a homogeneity map for the above "arrival confirmation entry."

101 means a menu screen, on which "arrival confirmation entry" can be selected.

102 is the W04 pallet for the screen (Fig. 4) on which to enter arrivals confirmation, whereas 103 is its W02 pallet.

104 is the W04 pallet for the warehouse codes reference screen (Fig. 5) at the time when "reference" of the "warehouse codes" has been selected in the screen on which

to enter arrival confirmation, whereas 105 is its W02 pallet.

When "execution" is selected in the screen on which to enter arrivals confirmation, it returns to the screen on which to enter arrivals confirmation.

When "registration" is selected in the screen on which to enter arrivals confirmation, the data is written onto a file via the WFL (Work File Area) 106.

When the "end" is selected in the screen on which to enter arrivals confirmation, it returns to the menu screen. Fig. 11 is a homogeneity map for the above "delivery request list."

111 is a menu screen on which "delivery request list" can be selected.

112 is the W04 pallet for the screen (Fig. 6) on which to output the delivery request list, whereas 113 is its W02 pallet.

When "print" is selected in the screen on which to output the delivery request list, the vouchers are output (printed) via the WFL (Work File Area) 114.

When "execution" is selected in the screen on which to output the delivery request list, it returns to the screen on which to output the delivery request list.

When "end" is selected in the screen on which to output the delivery request list, it returns to the menu screen.

In the homogeneity map, as explained above, for example, if a certain screen has been determined, the process to be followed (such as display of screen, output of voucher, writing onto file, reading from file, operation) will be

inevitably determined based on the contents of process of the screen (control words such as execution and registration), so that definitives such as screens are connected with lines in accordance with the above-mentioned contents of the process.

Determination of the file

Necessary files are determined based on the homogeneity map made and the words determined. That is, a necessary file can be inevitably determined from the homogeneity map and the word. A determined file is described in the appropriate place such as the dot-line box in the lower part outside the box of the homogeneity map.

For example, in the W02 pallet indicated by the code 93 of Fig. 9, it is consequently known, from the homogeneity map and the words, that the following files are necessary:

Employee M (master file)

Name M

Customer Information F (file)

Department M

W03-S/F

Product List F.

Creation of the Tense Control Vector

A tense control vector (program) for each word is to be made. That is, a tense control vector for each word is to be coded.

In the tense control vector, there are the following six kinds:

W04 duplicate vector

W04 homogeneity vector

W02 duplicate vector

W02 homogeneity vector

W03 duplicate vector

W03 homogeneity vector

These six kinds of the tense control vector (program) are to be created for every word. In some case, however, the W02 duplicate vector is not necessary.

For example, a sample for making the tense control vector is explained herein, focusing on the word "sales."

The W04 duplicate vector is to be made as shown in Fig. 12.

First, the vector should have the logic to make judgment whether a sales value exists or not in the address having the item "sales data" in the corresponding W02 of file (step 1201). If it exists, the vector should have the logic to set the sales value to the address having the item of "sales data" in the W04 (step 1202), thereby ending a process.

If the sales value does not exist there, the vector should have the logic to judge whether the sales value exists or not in the address having the item of "sales data" in the W03 (step 1203). If it exists, the vector should have

the logic to set the sales value to the address having the item of "sales data" in the W04 (step 1204), thereby ending a process. If the sales value does not exist there, the vector should have the logic to end the process without doing anything.

Furthermore, the vector should have a step, after the step 1204, to confirm if an aimed process is accomplished or not, and if the aimed process is not accomplished, to set a flag to reactivate the pallet.

What is important is that for any words, programs are to be made with always the same structure as the one stated above, which also applies to the following programs.

The W04 homogeneity vector is to be made as shown in Fig. 13.

First, the vector should have the logic to make judgment whether the sales value has already been input into the address having the item of "sales data" in the W04 (step 1301). If it has already been input, the vector should put an end to the process.

If the sales value has not been entered yet, the vector should have the logic to judge whether a "sales value" for the address having the item of "sales data" of the W04 can be computed or not by the internal data of the W04 (step 1302). That is, for example, assuming that the sales value is "the price" $\times$ "the number of pieces", the vector should judge whether values already exist or not in address having the item of "price" in the W04 and in the address having the item of "number of pieces" in the W04. If those values exist,

the vector should, deciding that it is computable, compute a "sales value" (for example, the vector is to compute a sales data from the existing "price" data and "number of pieces" data)(step 1303), and set the computed result (for example, the obtained "sales value") into the address having the item of "sales data" in the W04 (step 1304).

When the "sales data" is not computable, the vector should have the logic, if necessary (step 1305), set a flag for reactivating the pallet (step 1306). That is, the determination of the sales value is once reserved: for the above-mentioned example, the determination of the sales value is reserved until the "price" data and the "number of pieces" data are input. This means nothing but that the program stated in the present invention autonomously determines the significance.

The W02 homogeneity vector is to be made as shown in Fig. 14.

First, the vector should have the logic to make judgment whether a sales value is already input or not (that is, whether it is already set into the address having the item of the W04 "sales data" in the W04)(step 1401). If it is not input, the vector should put an end to the process.

If the sales value already input, the vector should have the logic to set the input sales value into the address having the item of "sales data" in the W02 (step 1402), and to put an end to the process.

For some words, if necessary, the vector should have,

before step 1401, a step for confirming whether the execution is necessary or not. Also, the vector should have, after step 1402, a step for confirming whether the aimed process is accomplished or not, and, if the aimed process has not been accomplished to set a flag for reactivating the pallet.

Further, for example in the case of the control words such as the "execute key," the above-mentioned step 1402 will become a process for setting a homogeneity map flag. The homogeneity map flag ($R = 0 - 5$, however, $R = 0$ is exceptional) is determined by the homogeneity map. By this homogeneity map flag it becomes possible to process distributing the homogeneity routes in the pallet chain function, as will be discussed later.

The W03 homogeneity vector is to be made as shown in Fig. 16.

First, the vector should have the logic to make judgment whether a sales value is already input or not into the address having the item of "sales data" in the W02 (step 1601). If it is already input, the vector should set the sales value into the address having the item of

the "sales data" in the W03 (step 1602), and to put an end to the process.

If the sales value does not exist, the vector should put an end to the process without doing anything.

For words, if necessary, the vector should have, before step 1601, a step for confirming whether the execution is necessary or not. Also, the vector should have, after step 1602, a step for confirming whether the aimed process is accomplished or not, and if the aimed process is not accomplished, to set a flag for reactivating the pallet.

The W03 homogeneity vector is to be made as shown in Fig. 17.

First, the vector should have the logic to make judgment whether a sales value is already input or not into the address having the item of "sales data" in the W03 (step 1701). If it is already input, the vector should put an end to the process.

If the sales value is not input, the vector should have the logic to make judgment whether the sales value for the address having the item of "sales data" in the W03 can be induced or not (step 1702). If it is not inducible, the vector should put an end to the process. If the sales value is inducible, the vector should have the logic to make judgment whether the calculation is possible or not (step 1703). That is, for example, assuming that the "sales value" is the "price" $\times$ the "number of pieces" as is the same as the step 1302 above, the vector should judge whether values already exist or not in the address having the item of the

"price" in the W03 and the "number of pieces" in the W03. Then, if it is inducible, the vector should calculate (for example, the vector should obtain the sales data from the existing "price" data and the "number of pieces" data) (step 1704), and to set the calculated result (for example, the obtained sales value) into the address having the item of the "sales data" in the W03 (step 1705).

When the calculation cannot be executed, if necessary (step 1706), the vector should set a flag for reactivating the pallet (step 1707). That is, also in this case, as the same as above, the determination of the sales value is once reserved: for the above-mentioned example, the determination of the sales value will be reserved until the "price" data and the "number of pieces" data are input. This means nothing but that the program stated in the present invention autonomously determines the significance.

Making of the pallet function

The pallet function for W04, W02 and W03 should be made. Fig. 18 shows the structure paradigm of the pallet function. The pallet function (program) having such structure may as well be made for each of W04, W02 and W03, as will be set forth in the followings.

First, the pallet function should have a logic to open a file (step 1801). That is, for example, the pallet function concerned with the W04 should have a logic to open a file of one predetermined W04 out of a plurality of W04 pallets, and then to execute each duplicate vector and homogeneity vector in the consecutive order (step 1802, step

1803).

Thereafter, in the case that the pallet activating flag has been set (step 1804), the function should reset the flag (step 1805), as well as execute each duplicate vector and homogeneity vector in the consecutive order (step 1802, step 1803). That is, the pallet function in co-operation with the pallet activating flag of each pallet, enables the determination of the autonomous significance of the program.

On the other hand, when the pallet activating flag has not been set, the function should close the file (step 1806) and put an end to the process.

Assembling into the Pallet Chain Function

The structure of the pallet chain function is identical in any software. Accordingly, the tense control vector and the pallet function made as explained above simply needs to be assembled into the pre-made pallet chain function.

The structure paradigm of the pallet chain function is shown in Fig. 19.

That is, the pallet chain function should have a logic to determine information to be transmitted (step 1901). The information to be transmitted is meant to be a screen to display. In the case of the homogeneity map shown in Fig. 9, for example, when the "reference" of the "customer code" in the sales data entry screen (ref. Fig. 2.) is selected (which means that a homogeneity map flag of $R = 3$ has been valid in the W02 pallet of "customer code"), the pallet chain function should determine the customer code reference screen (Fig. 3) as the information to be transmitted.

When the information to be transmitted contains the information for closing the system (step 1902), the function should have the logic to close the system (step 1903). The information for closing the system is that, explaining by referring to the homogeneity map shown in Fig. 9, for example, the "F3" has been selected in the sales data entry screen (Fig. 2) (which means that a homogeneity map flag of R=0 has been valid in the F3's W02 pallet). To close the system is, speaking of the above example, to return to the menu screen.

Next, the function should to make judgment whether to continue or not inside the WT (Walk-Through) unit (step 1904). This step is concerned with exceptional process. That is, the WT unit is, as shown in Fig. 20, for example, one unit made up by binding the entry screen used for data entry and the reference screen used for the reference to data upon entering the data. For example, by explaining in reference to the homogeneity map shown in Fig. 9, the WT unit is one unit made up by binding the sales data entry screen (ref. Fig. 2) and the customer code reference screen (ref. Fig. 3). However, when screens of other homogeneity maps, for example the arrival confirmation entry screen (ref. Fig. 10 and Fig. 4.), are needed, the confirmation entry screen must be unit-teamed with these screens stated above. Thereupon, as shown in Fig. 20, if necessary, the pallet chain should perform the scenario chain, or the chain with the other WT (step 1905, 1906). This will become particularly effective, when coping with a huge program.

In the case of the continuation inside the WT unit, the pallet chain function should set all pallets of the WT unit (step 1907). For example, in the case of the homogeneity map shown in Fig. 9, the function should set all the pallets concerned with this whole homogeneity map (W02 - W04) in the working memory, respectively.

Then, the function should execute the corresponding W04 pallet at first (step 1908). That is, the function should execute the W04 pallet in whose pallet function as shown in Fig. 18 all W04 duplicate vectors and W04 homogeneity vectors have been set. By this step, data concerned with the screen to display will be determined.

Then, the pallet chain function should execute the transmission function (step 1909). That is, the function should transmit a screen in which all the data have been set to a display means, for example, a CRT.

Thereafter, the function should have the logic to execute the receiving function (step 1910). That is, the function should receive the screen, in which all the data have been input (from display means, for example, a CRT).

Then, the pallet chain function should make judgment whether the received data is normal or not (step 1911). If abnormal, the function should resume the procedure from the beginning. That is, the function should make judgment if a data violating the regulation exists or not.

Next, the function should execute the corresponding W02 pallet (step 1912). That is, the function should execute the W02 pallet in whose pallet function as shown in Fig. 18 all

W02 homogeneity vectors have been set. By this step, the input data will be determined.

Next, a homogeneity map will be determined (step 1913, step 1914). As explained above, the homogeneity map is judged by the homogeneity map flag ($R = 1$ to 5) contained in the W02 homogeneity vector.

When the homogeneity map flag $R = 1$, the function should execute the corresponding W03 pallet (the W03 pallet which is not accompanied by a recording onto a file) (step 1915). For example, the line of the W03 pallet 94 shown in Fig. 9 is executed. Then, the function should have a logic to return to the first step (step 1901). That is, the homogeneity map in the case of $R = 1$ only performs data processing (ref. Fig. 9).

When the homogeneity map flag $R = 2$, the function should execute the corresponding W03 pallet (the pallet W03 accompanied by the recording onto a file) (step 1916). For example, the line of the W03 pallet 95 and the line of the WFL 96 are executed. Then the function should have a logic to return to the first process (step 1901). That is the homogeneity map in the case of $R = 2$ executes data processing as well as a recording data onto a file (ref. Fig. 9).

When the homogeneity map flag $R = 3$ to 5, the function should return to the first process (step 1901) as it is.

In this connection, the homogeneity map as shown by $R = 3$ performs processing to return to the W04 pallet (both homogeneous and heterogeneous) without doing anything

farther (ref. Fig. 9).

The homogeneity map as shown by $R = 4$ performs recording data in a file as it is (ref. Fig. 10).

The homogeneity map as shown by $R = 5$ performs taking out data from a file as it is (ref. Fig. 11).

To be noted, the pallet chain function shown in Fig. 19 is for on-line use. The pallet chain function for off-line use is as shown in Fig. 15.

In summary,

At first, the pallet chain function should have the logic to determine a screen to display (step 2101) and to activate the W04 pallet (step 2102). By this, screen data is edited and the edited screen is displayed (step 2103).

When the user's operation is done to this displayed screen, the function should have the logic to receive the screen (step 2104) and to activate the W02 pallet (step 2105). By this a homogeneity map route and the next screen to be displayed is judged.

Then, the function should have the logic to judge the homogeneity map (step 2106), and in the case of the homogeneity map 1 or 2 (step 2107), the function should activate the W03 pallet. By this, the data operation processing is done.

When the data operation processing is finished or in the case of other than the homogeneity map 1 or 2 above, the function returns to step 2101.

Fig. 22 shows the structure inside the pallet.

As shown by the Fig. 22, for the terms A - J on the screen 2201, duplicate vector A - J and homogeneity vector A - J exist as a pallet function inside the W04 pallet, as well as homogeneity vector A - J of the homogeneity vector PF1 and the homogeneity vector A - J of the homogeneity vector PFn exist as a pallet function inside the W02 pallet. Inside the W03 pallet, duplicate vector A - X and homogeneity vector A - X exist as a pallet function (A - X: all items).

A logic paradigm of the tense control vector is shown in Fig. 23. As shown in the Fig. 23, in the pallet chain function (scenario) 2301, when data is input and a screen/message is received, W02 pallet, W03 pallet and W04 pallet are activated in their order. When the W02 pallet is activated, the process shown by the code 2303 is executed. When the W03 pallet is activated, the process shown by the code 2303 is executed. When the W04 pallet is activated, the process shown by the code 2304 is executed.

Fig. 24 is a diagram showing characteristics of the logic in Lyee.

Suppose the program logic is Lc, and the logic being a base for causing actions in accordance with the information swept out by the program, which humans cannot be aware of but can instantly conduct is Lm, in the traditional way of making software, the logic Lm (impossible to be realized) being a base for causing human actions has been forced to be fabricated, a process (procedure) and a function both binding human actions have been fixed as specifications

(fabricated by SE with his or her experience or knowledge), thereby a program based thereupon has been made. That is, $L_c = L_m$

was its provision.

Lyee's software, contrary to the traditional method of making software, does not contain L_m , and the program to be made works in accordance with human's capricious conducts.

Fig. 25 and Fig. 26 show the structure of the traditional-type programs, and Fig. 27 shows the module configuration of the traditional-type programs.

As understood from these figures, the traditional-type programs can be referred to as process transaction of functional division type. For this reason, the structural condition is complicated, as well as the arrangement of the module configuration is also extremely difficult.

Fig. 28 shows the structure of the Lyee-type program.

As understood from this figure, the Lyee-type program features the followings: the configuration is simple; the element unit is processed by the unit of word; the element content is simple and independent; the program logic does not contain composition conditions (handling procedure) like traditional-type programs.

In the following, how to grasp business expertise and function as well as the resulting effects from the stand point of Lyee will be explained.

(1) There is no need for business expertise

Fig. 29 shows the W03 homogeneity vector.

In step 2901, the vector should have the logic to confirm whether the field's value is "space" or "zero". (judging from the item definition document). The judgment of this logic does not require business expertise; knowing the rule of Lyee is only needed.

In step 2902, the vector should have the logic to confirm whether the item in the right side (Starting point) of the corresponding item's expression (judging from the item definition document) exist or not in the W03 area. The judgment of this logic does not require business expertise; knowing the rule of Lyee is only required.

In step 2903, the vector should have the logic to confirm whether the value of the item of the right side (Starting point) of the corresponding item's expression (judging from the item definition document) is calculable ("space" or "zero") or not. The judgment of this logic does not require business expertise; knowing the rule of Lyee is only requested.

The step 2904 is a scheme which guarantees the operational order of the tense control vector. The judgment of this logic does not require business expertise; knowing the rule of Lyee is only necessary.

In step 2905, the vector shall calculate the items from the item definition document (such as expressions, etc.) users have confirmed.

Accordingly, in order to assemble Lyee's logic, the "item definition document such as expressions" is necessary, but the sequence of the process (which is called "business

expertise") is not required.

(2) Lyee deals with only synchronous data and dismiss a process (handling of asynchronous data).

In Lyee, by introducing the "duplicate vector," it is possible to handle only the synchronous data. That is, in Lyee, as shown in Fig. 30, there is no need of considering the examination of the logic for the operational procedure conditions, by handling only the synchronous data, whereas in the traditional methodology, much time was spent on examining the operational procedure, which results in worsening the productivity and the maintainanceability.

(3) The only one W03 pallet will do.

When realizing the logicalization of humans (i.e., to think communicable sentences, to take actions,....), it can be said from a biological point of view that all one's will and muscles (as represented by all brain cells) is intertwined each other. And, the right-side brain or the left-side brain primarily works in accordance with an object area to be logicalized. The logic of software (the logicalization), by nature, must be the same as this (because the software logic is nothing but the logic made by humans).

As shown in Fig. 31, the W03 pallet of Lyee is an area assuming the role of "logicalization," so that the W03 pallet installs all the words in a system, which is equivalent to all one's will and muscles, in one area as synchronous data. In the case of Lyee, tense control vectors corresponding to all words are all intertwined each other

(All tense control vectors work once, and as a result only the tense control vectors with significance execute their aimed process).

That is, traditionally, words (data) necessary for the function aimed at have to be examined and analyzed in advance before the data definition part is to be designed. On the contrary, in the case of Lyee, all needs to be done is that all words be defined as they are and that the individual tense control vectors be assembled into the logic as explained above.

(4) There are two kinds of the logic.

In the traditional software, the automatic logic was composed of the combination of an operational logic and an automatically self-run logic. On the other part, in the case of software made by Lyee method, as shown in Fig. 32, the automatically self-run logic and the operational logic are separated. That is, the automatically self-run logic exists in the W03 pallet, and the operational logic exists in the logic in a human brain.

(5) Checking

The traditional software, as shown in Fig. 33 (a), has been structured to be of a flow of checking against the input, performing a normal process in case of "OK", or performing an error transaction in case of "NG." Accordingly, all error conditions of multiple items must be examined beforehand.

On the other hand, in Lyee software, as shown in Fig. 33 (b), the only homogeneity map is determined without

executing the input check in the W02 pallet. In the W03 pallet, the selection is done concerning whether the process to be directed can be accomplished or not, and if the process to be directed cannot be executed, an error transaction is performed in the W04 pallet.

Explained next is the application range of Lyee in the system development step.

Fig. 34 (a) shows the traditional software development method. Traditionally, software has been developed through the following steps: scheme planning, job analysis and basic design, detail design, programming design, coding, and tests. Traditionally, development from the "job analysis and basic design" step to the "programming design" step has been performed depending upon SE's experience and expertise and human's abilities in coordinating meetings with users.

Fig. 34 (b) shows the Lyee's software development methodology. In Lyee, software is developed through the following steps: scheme planning, development into a homogeneity map, making of tense control vector specifications, coding of tense control vectors, and confirmation of movements. In Lyee, during the development process from the "development into a homogeneity map" step to "the confirmation of movements" step, the work is implemented plainly in accordance with the rule induced from the theory of Lyee, and the requirement which has been traditionally determined by human's abilities such as experience and expertise of SEs is solicited to enter in the

process of making users confirm the result, thereby bringing about the result that the system, by the users' responding to the soliciting, is to be made reality. So to speak, in the traditional method, the upper-stream requirement and the lower-stream source have been linked up each other by the SE's ability. On the other hand, in Lyee, the upper-stream requirement and the lower-stream source are linked by the theory of Lyee.

In the following, the working steps for the development of the Lyee-applied software is more specifically explained.

Fig. 35 (a) shows a hand-over document given by users to the development side. What are to be handed over to the development side includes screens, files, vouchers, and code tables. In this regard, as shown in Fig. 35 (b), the development side makes screen definition document, file definition document, voucher definition document, table of words, homogeneity maps, and tense control vectors specifications based on the discussion with users. Then, as shown in Fig. 35 (c),

the source program is made based on these documents.

Fig. 36 shows an example of the screen definition document. Fig. 37 shows an example of the voucher definition document. Fig. 38 shows an example of the file definition document. Fig. 39 shows an example of the homogeneity map.

Fig. 40 shows the method of making the homogeneity map.

(1) One shall confirm a screen for the menu to shift to at first, and make a framed space for the screen in the left-upper corner on the homogeneity map sheet.

(2) One shall make unconditionally a frame for W04 and W02 into this frame.

(3) One shall confirm the function key (button) on the screen, as well as distinguish between the screen to shift to upon pressing the key (button) and the function to be kept until the shift to the screen (*1: simple data hand-over/ *2: file making & updating/ *3: operation & confirmation without file making or updating has been completed).

(4) As the following example, one shall draw diverging branches in the right of W02, in accordance with the distinguishing result of the above (3).

*1: Simple data hand-over → Specifying the functional key, Homogeneity map (R) = 3, the frame of next screen's frame

*2: File making & updating → Specifying the functional key, Homogeneity map (R) = 2, the frame of W03

However, when a file to be made is other than the basic items, homogeneity map (R) = 4, the frame of WFL

*3: Simple operation & confirmation → Specifying the functional key, Homogeneity map $(R) = 1$, the frame of W03
 (5) Hereafter, one shall repeat the above (2) to (4) for all screens inside the unit of the homogeneity map.

In the following, the method of making the tense control vector specifications is explained based on Fig. 41. Fig. 41 indicates the homogeneity vector.

The condition for determining the "Input done" of step 4101 can be judged with the attribute and the initial value (known from the definitives definition document), and the others are all regulations in the course of realization by Lyee. Thus, no documents are required.

The condition for determining the "Inducibility" of step 4102 is whether the operational expression/editing expression is clear or not. Thus, no documents are required.

The condition for determining the "Operatability" of step 4103 is whether the value of the right side of the operational expression/editing expression may be used for the calculation or not, which is a condition that can be determined from the operational expression. Thus, no documents are required.

In step 4104, for example
 the following:

an expression like $Y = aX + bZ * cW$

and,

The range of possible values and particular conditions taken by a, b, c, X, Z, W

and,

The conditions to executing this computational formula (Multiple computational formulas are possible for one word), are to be questioned to users for each word and the answers are to be taken as a memo.

Fig. 42 and Fig. 43 show examples of the tense control vector specifications.

Fig. 44 shows an example of the programming based on the tense control vector specifications.

In this program, (1) to (4) are made from the rule of Lyee. That is, these are the particular conditions of Lyee, so interacting with users are not necessary.

(5) is to be made from the operational significant conditions of the tense control vector and the code table.

(6) is to be made from the operational expression of the tense control vector. (5) and (6) are to be determined and made into the documentation through interaction with the users, a hand-over document, "code table".

Hereinafter, the sample the program made by Lyee is illustrated.

	KH201E	Received Order Entry (Distribution)						DD/DD/DD TT:TT:TT	
1	O P C D	0000000000000000	Estimate No.	BBBBBBB	* Received Order Classification BB	00000000			
2	* Customer	BBBBBBBBB	00000000000000000000000000000000	TEL	000000000000				
3	* Delivery Destination	BBBBBBBBB	00000000000000000000000000000000	FAX	000000000000				
4	* Warehouse	BB	000000000000	* Person In Charge	BBBBBBB	00000000000000000000	Received Order No.	00000000	
5	Product Code	Product Name	Quantity	Previous Unit Price	Unit Price	Wholesale Rate	Delivery Deadline		
6	NO	Types & Media/Product #	Inventory	Price	Amount	Order No.			
7	Other Party Product								
8	1	BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	9999	6, 666, 666	9, 999, 999	BBB	999999		
9	0	00000000000000	BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	6, 666	6, 666, 666	666, 666, 666	BBBBBBBB		
10	2	BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	9999	6, 666, 666	9, 999, 999	BBB	999999		
11	0	00000000000000	BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	6, 666	6, 666, 666	666, 666, 666	BBBBBBBB		
12	3	BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	9999	6, 666, 666	9, 999, 999	BBB	999999		
13	0	00000000000000	BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	6, 666	6, 666, 666	666, 666, 666	BBBBBBBB		
14	4	BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	9999	6, 666, 666	9, 999, 999	BBB	999999		
15	0	00000000000000	BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	6, 666	6, 666, 666	666, 666, 666	BBBBBBBB		
16	5	BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	9999	6, 666, 666	9, 999, 999	BBB	999999		
17	0	00000000000000	BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	6, 666	6, 666, 666	666, 666, 666	BBBBBBBB		
18	6	BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	9999	6, 666, 666	9, 999, 999	BBB	999999		
19	0	00000000000000	BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	6, 666	6, 666, 666	666, 666, 666	BBBBBBBB		
20			Total Amount	6, 666, 666, 666					
21	Wholesale Cost No.	666							
22	Order-Originating Person	BBBBBBBBB	Mr./Mrs. Remarks	BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB	BBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBBB				
23	*Direct Shipment Classification B	Delivery Deadline	999999	Customer's Order Slip No.	BBBBBBBBB	* Dispatch	B	00000000000000	
24	F1- Registry	F1- Reference	F5- Credit Inquiry	F10/F11- Product Entry	(S/H)	F12- Cancel		F8	

2 - 1. HDR

PW1 V2R3MO 931105

SEU Origin List

File CTN010/QDDSSRC

Bar K052BP01

R*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+...

A*****

A* System Name = Catena New Sales/Distribution Management System

A* File Name = Receive Transaction File HDR

A* DDS Type = PF

A*

A* Entry Date = 95/ 8/24

A* Change Date = 95/ 9/05

A*****

A*

A R KD52RC

TEXT(' Receive Transaction File HDR R ')

A*

A KD1830 18A COLHDG(' IDX 1 Mgt. SN ')

A KD0325 8S 0 COLHDG(' Record Date ')

A KD0324 6S 0 COLHDG(' Record Time ')

A KDS208 1A COLHDG(' Delete Status ')

A KD1084 7A COLHDG(' Person In Charge Code ')

A KD1535 5A COLHDG(' Person In Charge Dept. Code ')

A KD1179 7A COLHDG(' Reg. Per. Code ')

A KD1181 5A COLHDG(' Reg. Per Dept. Code ')

A KDU975 18A COLHDG(' Change Orig. Mgt. SN ')

A KD1836 1A COLHDG(' HS Cl. ')

A KDU277 1A COLHDG(' Sales Cl. ')

A KD1838 1A COLHDG(' Kinds of Transaction ')

A KD1839 2A COLHDG(' Shipment Reason ')

A KD1840 2A COLHDG(' Arrival Reason ')

A KDS429 1A COLHDG(' Trns. F Cl. E ')

A KD1843 8S 0 COLHDG(' F Entry Date ')

A KD1844 8S 0 COLHDG(' Trns. No. ')

A KDC766 7A COLHDG(' BS Report No. ')

A KD1847 9A COLHDG(' Customer Code ')

A KD1848 220 COLHDG(' Name of Customer Person In Charge ')

A KD4044 20A COLHDG(' Name (Kana) of Customer Person In Charge ')

A KD4000 420 COLHDG(' Trns. Name ')

A KD1849 280 COLHDG(' Trns. Valid Period ')

A KD1850 2S 0 COLHDG(' No. of TXT Lines ')

A KD1851 8S 0 COLHDG(' Trns. End Date Sched. ')

A KD1852 18A COLHDG(' HDR Mgt. SN ')

A KD1854 13P 0 COLHDG(' Trns. Total Amount ')

A KD1855 13P 0 COLHDG(' Trns. Total Amt. inclusive of Consumption Tax ')

A KD1856 13P 0 COLHDG(' Qty Pr. Total Amt. ')

A KD1857 13P 0 COLHDG(' Qty Unit Price Total Amt. ')

A KD1858 13P 0 COLHDG(' Qty Unit Price Disc. Total Amt. ')

A KD1859 13P 0 COLHDG(' C. Tax Total Amt. ')

A KDU360 1A COLHDG(' Disc. Cl. ')

A KDB001 4S 2 COLHDG(' Sales Unit Price Disc. Rate ')

A KD1099 13P 0 COLHDG(' Disc. Total Amt. ')

A KD1865 9P 2 COLHDG(' Total No. of Prod. Transacted ')

A KDU470 7P 0 COLHDG(' Total No. of Prod. Untransacted ')

A KD1866 7P 0 COLHDG(' No. of Possible Security Prod. ')

A KD1867 7P 0 COLHDG(' No. of Impossible Security Prod. ')

A KD0439 7S 0 COLHDG(' Settlement Bank Code ')

A KD1877 1S 0 COLHDG(' Credit Sales Acceptance FLG ')

A KD0019 13P 0 COLHDG(' Credit Sales Total Amt. ')

PW1 V2R3MO 931105

SEU Origin List

File CTN010/QDDSSRC

Bar K052BP01

R#	1	2	3	4	5	6	7
A	KD0020	13P 0		COLHDG (' Credit Sales Paid-in Amt.			
A	KD0021	8S 0		COLHDG (' Credit Sales Strike-out Date			
A	KD0022	13P 0		COLHDG (' Credit Sales Not Striked-out			
A	KD1878	1S 0		COLHDG (' Credit Purchase Acpt. FLG			
A	KD0023	13P 0		COLHDG (' Credit Purchase Total Amt.			
A	KD0024	13P 0		COLHDG (' Credit Purchase Payment Amt.			
A	KD0025	8S 0		COLHDG (' Credit Purchase Strike-out Date			
A	KD0026	13P 0		COLHDG (' Credit Purchase Strike-out Remainder			
A	KD1412	1A		COLHDG (' Dispatch Cl.			
A	KD4045	2A		COLHDG (' Dispatcher Code			
A	KD4046	1A		COLHDG (' Group Packing Cl. No.			
A	KD0059	1A		COLHDG (' Delivery Note Issuance FLG			
A	KDS228	1A		COLHDG (' KH3010 Issuance Instrn. FLG			
A	KDS229	1A		COLHDG (' KG3020 Issuance Instrn. FLG			
A	KDS242	1A		COLHDG (' KH4090 Issuance Instrn. FLG			
A	KDS284	1A		COLHDG (' KH5260 Issuance Instrn. FLG			
A	KDS398	1A		COLHDG (' Reserve 1			
A	KDS399	1A		COLHDG (' Reserve 2			
A	KDS400	1A		COLHDG (' Reserve 3			
A	KDS401	1A		COLHDG (' Reserve 4			
A	KDS402	1A		COLHDG (' Order Acpt. Center Person In Charge Cl.			
A**							
A	KDS403	1A		COLHDG (' Change Screen Cl.			
A**							
A**							
A**							
A**							
A	KDS404	1A		COLHDG (' Reserve 7.			
A	KDS405	1A		COLHDG (' Acct. I/F Done FLG			
A	KDS406	1A		COLHDG (' Reverse G-NG Cl.			
A	KDS407	1A		COLHDG (' Over Credit Limit Amt. FLG			
A	KDS408	1A		COLHDG (' Next Trns. Shift FLG			
A	KDS409	1A		COLHDG (' Trns. Correction Done FLG			
A	KD4028	8S 0		COLHDG (' Estimate No.			
A	KD4029	8S 0		COLHDG (' Rec. Order No.			
A	KDU452	2S 0		COLHDG (' Rec. Order Line No.			
A	KD4030	8S 0		COLHDG (' Placed Order No.			
A	KDU453	2S 0		COLHDG (' Placed Order Line No.			
A	KD4031	8S 0		COLHDG (' Buying-in No.			
A	KDU454	2S 0		COLHDG (' Buying-in Line No.			
A	KD4032	8S 0		COLHDG (' Shipping No.			
A	KDU455	2S 0		COLHDG (' Ship. Line No.			
A	KD4033	8S 0		COLHDG (' Payment No.			
A	KD4034	8S 0		COLHDG (' Paid-in No.			
A	KD4035	8S 0		COLHDG (' Sales No.			
A	KDU456	2S 0		COLHDG (' Sales Line No.			
A	KD4036	8S 0		COLHDG (' Move No.			
A	KDU457	2S 0		COLHDG (' Move Line No.			
A	KDU289	8S 0		COLHDG (' Sales Return Appln. No.			
A	KDU458	2S 0		COLHDG (' Sales Return Appln. Line			
A	KDU290	8S 0		COLHDG (' Buying-in Return Appln. No.			
A	KDU459	2S 0		COLHDG (' Buying-in Return Appln. Line			
A	KDS563	8S 0		COLHDG (' Resell No.			

PW1 V2R3MO 931105

SEU Origin List

File CTN010/QDDSSRC

Bar K052BP01

R*	1	2	3	4	5	6	7	8
A		KD1057	8S	0	COLHDG	(	Loan No.	)
A		KDU936	2S	0	COLHDG	(	Loan Line No.	)
A		KDU937	8S	0	COLHDG	(	Loan Return No.	)
A		KDU938	2S	0	COLHDG	(	Loan Return Line No.	)
A		KD0944	7A		COLHDG	(	Bill No.	)
A		KD0511	2A		COLHDG	(	Account Cl.	)
A		KD0512	7A		COLHDG	(	Account #	)
A		KD1786	420		COLHDG	(	Remarks 1	)
A		KD1787	420		COLHDG	(	Remarks 2	)
A		KDS410	420		COLHDG	(	Remarks 3	)
A		KDS411	420		COLHDG	(	Remarks 4	)
A		KDS412	420		COLHDG	(	Remarks 5	)
A		KD4037	200		COLHDG	(	Customer Trns. No.	)
A		KDS564	1A		COLHDG	(	Customer Slips Genuine/Temp. Cl.	)
A		KDS565	1A		COLHDG	(	Inventory Security Done FLG	)
A		KD0959	9A		COLHDG	(	Billing Code	)
A		KD1571	2A		COLHDG	(	Return Reason Code	)
A		KD1447	2A		COLHDG	(	Sales Cl.	)
A		KDU309	1A		COLHDG	(	Consumption Tax Y/N Cl.	)
A		KDU364	1A		COLHDG	(	Sales Sum-up Cl.	)
A		KD0201	1A		COLHDG	(	Install Cl.	)
A		KD0692	1A		COLHDG	(	Auto. Shipping Cl.	)
A		KDS566	1A		COLHDG	(	Partial Delivery Possibility Cl.	)
A		KD0121	6A		COLHDG	(	IBM Customer Code	)
A		KDS567	1A		COLHDG	(	Maker's Direct Distribution Cl.	)
A		KDS568	1A		COLHDG	(	Catena Direct Distribution Cl.	)
A		KDS027	8A		COLHDG	(	Acct. Customer Code	)
A		KD0654	1A		COLHDG	(	Designated Slip Cl.	)
A		KDV201	11A		COLHDG	(	Customer Designated Trns. No.	)
A		KD1261	10A		COLHDG	(	Customer name (kana)	)
A		KDV320	1A		COLHDG	(	EOS Cl.	)
A		KDV321	1A		COLHDG	(	Reserve Cl. 2	)
A		KD1771	10A		COLHDG	(	Delivery Destination Code	)
A		KD1772	10A		COLHDG	(	Reserve 2	)
A		KD1773	10A		COLHDG	(	Reserve 3	)
A*								
A		K KD1852						
A*								

* * * End of Origin Specification * * *

2 - 2. D T L

PW1 V2R3MO 931105

SEU Origin List

File CTN010/QDDSSRC

Bar K052BP02

R*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+...

A*****
 A* System Name = Catena New Sales/Distribution Management System
 A* File Name = Receive Transaction File DTL
 A* DDS Type PF = PF
 A*
 A* Entry Date = 95/ 8/30
 A* Change Date = 95/ 8/30
 A*****

A*	UNIQUE	TEXT('	Receive Transaction File DTL	')
A	R KB52RC			
A*				
A	KB4001	18A	COLHDG('	TXT Mgt. SN (Serial No.) ')
A	KB1879	2S 0	COLHDG('	TXT No. ')
A	KES208	1A	COLHDG('	Reserve Cl. (delete) ')
A	KB1084	7A	COLHDG('	Person In Charge Code ')
A	KB1535	5A	COLHDG('	Person In Charge Dept. Code ')
A	KEU277	1A	COLHDG('	Sales Cl. ')
A	KB1838	1A	COLHDG('	Kinds of Transaction ')
A	KB1839	2A	COLHDG('	Shipment Reason ')
A	KB1840	2A	COLHDG('	Arrival Reason ')
A	KB1843	8S 0	COLHDG('	F Entry Date ')
A	KB1844	8S 0	COLHDG('	Tms. No. ')
A	KB1847	9A	COLHDG('	Customer Code ')
A	KB1447	2A	COLHDG('	Sales Cl. ')
A	KES567	1A	COLHDG('	Maker's Direct Distribution Cl. ')
A	KES568	1A	COLHDG('	Catena Direct Distribution Cl. ')
A	KB1836	1A	COLHDG('	HS Cl. ')
A	KB4101	12A	COLHDG('	Product Code 1 ')
A	KEQ027	600	COLHDG('	Product Code Product Name ')
A	KE0130	13A	COLHDG('	JAN Code ')
A	KB4301	260	COLHDG('	Product Name (Abbrev.) 1 ')
A	KB4351	12A	COLHDG('	Type Name (Abbrev.) 1 ')
A	KB4401	10A	COLHDG('	Media Name 1 ')
A	KB4451	15A	COLHDG('	Product No. 1 ')
A	KB4042	13A	COLHDG('	Customer Product Code ')
A	KB5001	7P 2	COLHDG('	No. of Transacted Products ')
A	KES413	5P 0	COLHDG('	Estimated No. of Transacted Products ')
A	KEU471	5P 0	COLHDG('	Balance of Transacted Products ')
A	KB5351	1S 0	COLHDG('	Freq. of Security ')
A	KB5051	200	COLHDG('	Customer Tms. No. ')
A	KB0747	1A	COLHDG('	Agency Cl. ')
A	KB5101	1A	COLHDG('	Inventory Cl. 1 ')
A	KB5151	9A	COLHDG('	Inventory Location Code 1 ')
A	KB5201	1A	COLHDG('	Security Method 1 ')
A	KE1070	5A	COLHDG('	Shelf No. 1 ')
A	KB5251	1A	COLHDG('	Product Status Cl. 1 ')
A	KB5301	7P 0	COLHDG('	Nm. of Inventory Location Stock 1 ')
A	KBS203	7P 0	COLHDG('	Inventory Location Security No. 1 ')
A	KB5102	1A	COLHDG('	Inventory Cl. 2 ')
A	KD5152	9P	COLHDG('	Inventory Location Code 2 ')
A	KD5202	1A	COLHDG('	Security Method 2 ')
A	KD1071	5A	COLHDG('	Shelf No. 2 ')

PW1 V2R3MO 931105

SEU Origin List

File CTN010/QDDSSRC

Bar K052BP02

R#	1	2	3	4	5	6	7
A		KE5252	1A		COLHDG (' Product Status Cl. 2		
A		KE5302	7P 0		COLHDG (' Nm. of Inventory Location Stock 2		
A		KE5204	7P 0		COLHDG (' Inventory Location Security No. 2		
A		KE5103	1A		COLHDG (' Inventory Cl. 3		
A		KE5153	9A		COLHDG (' Inventory Location Code 3		
A		KE5203	1A		COLHDG (' Security Method 3		
A		KE1072	5A		COLHDG (' Shelf No. 3		
A		KE5253	1A		COLHDG (' Product Status Cl. 3		
A		KE5303	7P 0		COLHDG (' Nm. of Inventory Location Stock 3		
A		KE5205	7P 0		COLHDG (' Inventory Location Security No. 3		
A		KE5104	1A		COLHDG (' Inventory Cl. 4		
A		KE5154	9A		COLHDG (' Inventory Location Code 4		
A		KE5204	1A		COLHDG (' Security Method 4		
A		KE1073	5A		COLHDG (' Shelf No. 4		
A		KE5254	1A		COLHDG (' Product Status Cl. 4		
A		KE5304	7P 0		COLHDG (' Nm. of Inventory Location Stock 4		
A		KE5206	7P 0		COLHDG (' Inventory Location Security No. 4		
A		KE5105	1A		COLHDG (' Inventory Cl. 5		
A		KE5155	9A		COLHDG (' Inventory Location Code 5		
A		KE5205	1A		COLHDG (' Security Method 5		
A		KE5428	5A		COLHDG (' Shelf No. 5		
A		KE5255	1A		COLHDG (' Product Status Cl. 5		
A		KE5305	7P 0		COLHDG (' Nm. of Inventory Location Stock 5		
A		KE5207	7P 0		COLHDG (' Inventory Location Security No. 5		
A		KE1904	7P 0		COLHDG (' Product Security Total		
A		KEU480	620		COLHDG (' Final Arrival Point Name 1 1		
A		KE5551	8A		COLHDG (' Final Arrival Point Postal Code 1		
A		KE5601	420		COLHDG (' Final Arrival Point Address 1-1		
A		KE5651	420		COLHDG (' Final Arrival Point Address 2-1		
A		KE5701	420		COLHDG (' Final Arrival Point Address 3-1		
A		KE5751	320		COLHDG (' Final Arrival Point Department 1		
A		KE5801	220		COLHDG (' Final Arrival Point Person in Charge 1		
A		KE5851	12A		COLHDG (' Final Arrival Point TEL 1		
A		KE5901	12A		COLHDG (' Final Arrival Point FAX 1		
A		KE9101	5P 0		COLHDG (' Nm. of Arrival Products at Arrival-point 1		
A		KE5401	9A		COLHDG (' Start Point Location Code 1		
A		KE5451	9A		COLHDG (' Arrival Point Location Code 1		
A		KE5501	9A		COLHDG (' Final Arrival Point Code 1		
A		KE5951	8S 0		COLHDG (' Arrival-point Arrival Schedule 1		
A		KE6001	8S 0		COLHDG (' Final Arrival-point Arrival Schedule 1		
A		KE6051	1A		COLHDG (' Routing Judgement 1		
A		KE5569	1A		COLHDG (' Early/Mid/Late of Month Schl. Cl. 1		
A		KEU460	8S 0		COLHDG (' Tms. Done No.1 1		
A		KEU461	2S 0		COLHDG (' Tms. Done Line No.1		
A		KE9351	5P 0		COLHDG (' Nm. of Arrival Products at Arrival-point 2		
A		KE6102	9A		COLHDG (' Start Point Location Code 2		
A		KE6152	9A		COLHDG (' Arrival Point Location Code 2		
A		KE6202	9A		COLHDG (' Final Arrival Point Code 2		
A		KE6652	8S 0		COLHDG (' Arrival-point Arrival Schedule 2		
A		KE6702	8S 0		COLHDG (' Final Arrival-point Arrival Schedule 2		
A		KE6752	1A		COLHDG (' Routing Judgement 2		
A		KE5570	1A		COLHDG (' Early/Mid/Late of Month Schl. Cl. 2		
A		KEU462	8S 0		COLHDG (' Tms. Done No.2 1		

PW1 V2R3MO 931105

SEU Origin List

File CTN010/QDDSSRC

Bar K052BP02

R*	1	2	3	4	5	6	7	8
A	KEU463	2S	0	COLHDG	(	Tms. Done Line No.2	)	
A	KE9601	5P	0	COLHDG	(	Nm. of Arrival Products at Arrival-Point 3	)	
A	KE6103	9A		COLHDG	(	Start Point Location Code 3	)	
A	KE6153	9A		COLHDG	(	Arrival Point Location Code 3	)	
A	KE6903	9A		COLHDG	(	Final Arrival Point Code 3	)	
A	KE7353	8S	0	COLHDG	(	Arrival-point Arrival Schedule 3	)	
A	KE7403	8S	0	COLHDG	(	Final Arrival-point Arrival Schedule 3	)	
A	KE7453	1A		COLHDG	(	Routing Judgement 3	)	
A	KES571	1A		COLHDG	(	Early/Mid/Late of Month Schl. Cl. 3	)	
A	KEU464	8S	0	COLHDG	(	Tms. Done No.3 1	)	
A	KEU465	2S	0	COLHDG	(	Tms. Done Line No.3	)	
A	KE9851	5P	0	COLHDG	(	Nm. of Arrival Products at Arrival-Point 4	)	
A	KE7504	9A		COLHDG	(	Start Point Location Code 4	)	
A	KE7554	9A		COLHDG	(	Arrival Point Location Code 4	)	
A	KE7604	9A		COLHDG	(	Final Arrival Point Code 4	)	
A	KE8054	8S	0	COLHDG	(	Arrival-point Arrival Schedule 4	)	
A	KE8104	8S	0	COLHDG	(	Final Arrival-point Arrival Schedule 4	)	
A	KE8154	1A		COLHDG	(	Routing Judgement 4	)	
A	KES572	1A		COLHDG	(	Early/Mid/Late of Month Schl. Cl. 4	)	
A	KEU466	8S	0	COLHDG	(	Tms. Done No.4 1	)	
A	KEU467	2S	0	COLHDG	(	Tms. Done Line No.4	)	
A	KEA101	5P	0	COLHDG	(	Nm. of Arrival Products at Arrival-Point 5	)	
A	KE8205	9A		COLHDG	(	Start Point Location Code 5	)	
A	KE8255	9A		COLHDG	(	Arrival Point Location Code 5	)	
A	KE8305	9A		COLHDG	(	Final Arrival Point Code 5	)	
A	KE8755	8S	0	COLHDG	(	Arrival-point Arrival Schedule 5	)	
A	KE8805	8S	0	COLHDG	(	Final Arrival-point Arrival Schedule 5	)	
A	KE8855	1A		COLHDG	(	Routing Judgement 5	)	
A	KES573	1A		COLHDG	(	Early/Mid/Late of Month Schl. Cl. 5	)	
A	KEU468	8S	0	COLHDG	(	Tms. Done No.5 1	)	
A	KEU469	2S	0	COLHDG	(	Tms. Done Line No. 5	)	
A	KE4005	7P	0	COLHDG	(	Procurement Impossible Total	)	
A	KE1124	13P	3	COLHDG	(	Regular Price	)	
A	KEA651	13P	3	COLHDG	(	Manufacturing Unit Price	)	
A	KEA701	4S	2	COLHDG	(	Manufacturing Unit Price Discount Rate	)	
A	KEA751	4S	2	COLHDG	(	Manufacturing Special Discount Rate	)	
A	KEA801	13P	3	COLHDG	(	Buying-in Unit Price	)	
A	KEU472	9P	0	COLHDG	(	Buying-in Unit Price Discount Amount	)	
A	KEA851	4S	2	COLHDG	(	Buying-in Unit Price Discount Rate	)	
A	KEA901	4S	2	COLHDG	(	Buying-in Special Discount Rate	)	
A	KEA951	13P	3	COLHDG	(	Sales Unit Price	)	
A	KEU473	9P	0	COLHDG	(	Sales Unit Price Discount Amount	)	
A	KEB001	4S	2	COLHDG	(	Sales Unit Price Discount Rate	)	
A	KEB051	4S	2	COLHDG	(	Sales Special Discount Rate	)	
A	KEB701	13P	3	COLHDG	(	Loan Unit Price	)	
A	KEU478	9P	0	COLHDG	(	Loan Unit Price Discount Amount	)	
A	KEB751	4S	2	COLHDG	(	Loan Unit Price Discount Rate	)	
A	KEB801	4S	2	COLHDG	(	Loan Special Discount Rate	)	
A	KEC001	9P	0	COLHDG	(	Sales/Buying-in Rough Profit	)	
A	KEC151	5S	2	COLHDG	(	Sales/Buying-in Rough Profit Rate	)	
A	KE0373	3S	0	COLHDG	(	Wholesale Rate	)	
A	KEC201	9P	0	COLHDG	(	Transacted Quantity Unit Price Amount	)	
A	KEC251	9P	0	COLHDG	(	Transacted Quantity Unit Price Discount Amount	)	

PW1 V2R3MO 931105

SEU Origin List

File CTN010/QDDSSRC

Bar K052BP02

R*...+... 1 ...+... 2 ...+... 3 ...+... 4 ...+... 5 ...+... 6 ...+... 7 ...+...

A	KES002	9P 0	COLHDG (' Transaction on Consumption tax Amount)
A	KB4028	8S 0	COLHDG (' Estimate No.
A	KB4029	8S 0	COLHDG (' Rec. Order No.
A	KEU452	2S 0	COLHDG (' Rec. Order Line No.
A	KE4030	8S 0	COLHDG (' Placed Order No.
A	KEU453	2S 0	COLHDG (' Placed Order Line No.
A	KB4031	8S 0	COLHDG (' Buying-in No.
A	KEU454	2S 0	COLHDG (' Buying-in Line No.
A	KE4032	8S 0	COLHDG (' Shipping No.
A	KEU455	2S 0	COLHDG (' Ship. Line No.
A	KB4033	8S 0	COLHDG (' Payment No.
A	KE4034	8S 0	COLHDG (' Paid-in No.
A	KB4035	8S 0	COLHDG (' Sales No.
A	KEU456	2S 0	COLHDG (' Sales Line No.
A	KE4036	8S 0	COLHDG (' Move No.
A	KEU457	2S 0	COLHDG (' Move Line No.
A	KEU289	8S 0	COLHDG (' Sales Return Appln. No.
A	KEU453	2S 0	COLHDG (' Sales Return Appln. Line 0
A	KEU290	8S 0	COLHDG (' Buying-in Return Appln. No.
A	KEU459	2S 0	COLHDG (' Buying-in Return Appln. Line 0
A	KBS563	8S 0	COLHDG (' Resell No.
A	KE1057	8S 0	COLHDG (' Loan No.
A	KEU936	2S 0	COLHDG (' Loan Line No.
A	KEU937	8S 0	COLHDG (' Loan Return No.
A	KEU938	2S 0	COLHDG (' Loan Return Line No.
A	KE0944	7A	COLHDG (' Bill No.
A	KE1571	2A	COLHDG (' Reason Code
A	KEU529	1A	COLHDG (' Payment Hold Cl.
A	KET013	40A	COLHDG (' Product Name (kana)
A	KET030	9A	COLHDG (' Reserve A
A	KET031	9A	COLHDG (' Reserve B
A	KET032	9A	COLHDG (' Reserve C
A	KET033	9A	COLHDG (' Reserve D
A	KET034	9A	COLHDG (' Reserve E
A	KET035	9A	COLHDG (' Reserve F
A	KET036	9A	COLHDG (' Reserve G
A	KET037	9A	COLHDG (' Registry-Time Mgt. SN Up 9
A	KET038	9A	COLHDG (' Registry-Time Mgt. SN Down 9
A	KET039	9A	COLHDG (' Product Buying-in Party Code
A	KEV320	1A	COLHDG (' EOS Cl.
A	KEB321	1A	COLHDG (' Reserve Cl. 2
A	KB1771	7A	COLHDG (' Reserve 1
A*			
A	K KB4001		
A	K KB1879		
A*			

* * * End of Origin Specification * * *

W02 Homogeneity Vector (Reference Key)

 * LP0004 P F 4 : Reference

LP0004-SEC SECTION.

LP0004-START
 IF NOT (PRKEY NO OF W0204(PS) = 04)
 GO TO LP0004-BXIT
 END-IF.

 * L01540 O P C D (LP0004)

IF ERR-INF NOT = "1"
 IF A01540 OF KH201E(PS) = SPACE
 MOVE "1" TO ERR-INF
 MOVE 2009 TO MSG-NO
 MOVE B"1" TO A01540-C OF W0204(PS)
 MOVE SPACE TO A01560 OF KH201E(PS)
 ELSE

* OPEN K213DL01

IF K213DL01-OPEN NOT = "1"
 OPEN INPUT K213DL01
 IF FLSTS1 NOT = ZERO
 MOVE 8800 TO MSG-NO
 MOVE "1" TO ERR-INF
 GO TO MAIN-END
 ELSE
 MOVE "1" TO K213DL01-OPEN
 END-IF
 END-IF

*

MOVE A01540 OF KH201E(PS) TO W01540
 MOVE ZERO TO SGLCODE
 MOVE W01540 TO SK1084 OF K213DL01-REC
 READ K213DL01
 INVALID KEY
 MOVE 100 TO SGLCODE
 END-READ
 IF SGLCODE = ZERO
 MOVE SKD792 OF K213DL01-REC TO R-W01560
 END-IF

*-----

IF SGLCODE = ZERO
 MOVE R-W01560 TO INDATA
 MOVE 16 TO OUTKETA
 PERFORM PSLEKPROC
 MOVE OUTDATA TO A01560 OF KH201E(PS)
 ELSE
 MOVE "1" TO ERR-INF
 MOVE 2001 TO MSG-NO
 MOVE B"1" TO A01540-C OF W0204(PS)
 MOVE SPACE TO A01560 OF KH201E(PS)
 END-IF
 END-IF

* BND-IF
 * Determination of homogeneity route and the next screen in the case of customer code

IF ERR-INF NOT = "1"
 IF CSR-LOCK OF W0204(PS) = "A12590"
 MOVE 3 TO ROUTE-NO
 MOVE "KH600Q" TO L-GAMEN-NO
 BND-IF

* Determination of homogeneity route and the next screen in the case of Delivery destination code

IF CSR-LOCK OF W0204(PS) = "A13610"
 MOVE 3 TO ROUTE-NO
 MOVE "KH600Q" TO L-GAMEN-NO

W03: Duplicate Vector

```

***** Beginning of Data *****
*****
***** KH201E Received Order Entry (Distribution) *****
***** IF GAMEN-NO OF CRTL-AREA = "KH201E" *****
***** DISPLAY "KH201E LAYT-FLG=" LAYH-FLG *****
D  DISPLAY "KH201E LAYT-FLG=" LAYT-FLG
D  IF LAYH-FLG = 1
***** Reg. Per. Code YD1179KH201E1E
      MOVE A01540 OF KH201E(STG1)
      TO KD1179 OF EF-AREA
***** Shipment Reason YD1839KH201E1E
      MOVE A07580 OF KH201E(STG1)
      TO KD1839 OF EF-AREA
***** Customer Code YD1847KH201E1E
      MOVE A12590 OF KH201E(STG1)
      TO KD1847 OF EF-AREA
960424 ***** Transaction No. YD1844KH201E1E
***** IF A07560 OF KH201E(STG1) IS NUMERIC
***** MOVE A07560 OF KH201E(STG1)
***** TO KD1844 OF EF-AREA
***** END-IF
960513 ***** Received Order No. YD1844KH201E1E
***** IF A07560 OF KH201E(STG1) IS NUMERIC
***** MOVE A07560 OF KH201E(STG1)
***** TO KD4029 OF EF-AREA
***** EDN-IF
960513 ***** Estimate No. YD4028KH201E1E
***** IF A04800 OF KH201E(STG1) IS NUMERIC
***** MOVE A04800 OF RH201E(STG1)
***** TO KD4028 OF EF-AREA
***** END-IF
***** Trns. Total Amt. YD1854KH201E1E
      MOVE A04360 OF KH201E(STG1)
      TO KD1854 OF EF-AREA
***** Qty Unit Price Total Amt. YD1857KH201E1E
      MOVE A04360 OF KH201E(STG1)
      TO KD1857 OF EF-AREA
***** Name of Customer Person In Charge (Tentative) YD4044KH201E1E
      MOVE A14890 OF KH201E(STG1)
      TO KD4044 OF EF-AREA
***** Remarks 1 YD1786KH201E1E
      MOVE A17860 OF KH201E(STG1)
      TO KD1786 OF EF-AREA
***** Remarks 2 YD1787KH201E1E
      MOVE A17870 OF KH201E(STG1)
      TO KD1787 OF EF-AREA
***** Customer Trns. No. YD4037KH201E1E
      MOVE AU2980 OF KH201E(STG1)
      TO KD4037 OF EF-AREA
***** Distribution Cl. YD1412KH201E1E
      MOVE A14130 OF KH201E(STG1)
      TO KD1412 OF EF-AREA
***** Person In Charge Code YD1084KH201E1E
      MOVE A10840 OF KH201E(STG1)
      TO KD1084 OF EF-AREA
***** Catena Direct Distribution Cl. YDS568KH201E1E
      MOVE A11210 OF KH201E(STG1)
      TO KDS568 OF EF-AREA
      END-IF
***** ***** TEXT
      IF LAYT-FLG = "1"
      PERFORM VARYING SI FROM 1 BY 1
      UNTIL SI > 6
***** LASTIDX CONTROL
      MOVE SI TO TI
***** Final Arrival Point Code 1 YB5501KH201E1E
***** IF A11210 OF KH201E(STG1) NOT = 1

```

W03: Homogeneity Vector

```

***** Beginning of Data ***** W03BH2 *****
***** Received Order (HDR) *****
***** Quantity Unit Price Total Amount L18570 *****
UG0717***** IF TKBN = CNS-BF AND KDU360 OF EF-AREA NOT = "1"
               IF TKBN = CNS-BF
                   IF KE5001 OF EF-AREA(PI) IS NUMERIC
                       AND KD1857 OF EF-AREA IS NUMERIC
                       AND KEC201 OF EF-AREA(PI) IS NUMERIC
                           IF PI = 1
                               MOVE ZERO TO KD1857 OF EF-AREA
                           END-IF
                           IF KE5001 OF EF-AREA(PI) NOT = ZERO
                               COMPUTE DK1857 OF EF-AREA = KD1857 OF EF-AREA
                                                                + KEC201 OF EF-AREA(PI)
                               ON SIZE ERROR
                                   MOVE ALL "9" TO KD1857 OF EF-AREA
                                   IF ERR-INF = SPACE
                                       MOVE 1 TO ERR-INF
                                       MOVE 3019 TO MSG-NO
                                   END-IF
                               END-COMPUTE
                           END-IF
                       END-IF
                   END-IF
               ***** Quantity Unit Price discount Total Amount L18580 *****
               IF TKBN = CNS-BF
                   IF KD1858 OF EF-AREA IS NUMERIC
                       AND KE5001 OF EF-AREA(PI) IS NUMERIC
                       AND KEC251 OF EF-AREA(PI) IS NUMERIC
                           ***** Discount Cl. NOT = "1" *****
                           IF KDU360 OF EF-AREA NOT = "1"
                               IF PI = 1
                                   MOVE ZERO TO KD1858 OF EF-AREA
                               END-IF
                               IF KE5001 OF EF-AREA(PI) NOT = ZERO
                                   COMPUTE KD1858 OF EF-AREA
                                       = KD1858 OF EF-AREA
                                       + KEC251 OF EF-AREA(PI)
                                   ON SIZE ERROR
                                       MOVE ALL "9" TO KD1858 OF EF-AREA
                                       IF ERR-INF = SPACE
                                           MOVE 1 TO ERR-INF
                                           MOVE 3019 TO MSG-NO
                                       END-IF
                                   END-COMPUTE
                               END-IF
                           END-IF
                       END-IF
                   ***** Discount Cl. = "1" *****
                   IF KDU360 OF EF-AREA = "1"
                       MOVE KD1099 OF EF-AREA
                           TO KD1858 OF EF-AREA
                   END-IF
               END-IF
               ***** Quantity Regular Price Total Amount L18560 *****
               IF TKBN = CNS-BF
                   IF KE5001 OF EF-AREA(PI) IS NUMERIC
                       AND KD1856 OF EF-AREA IS NUMERIC
                       AND KE1124 OF EF-AREA(PI) IS NUMERIC
                           IF PI = 1
                               MOVE ZERO TO KD1856 OF EF-AREA
                           END-IF
                           IF KE5001 OF EF-AREA(PI) NOT = ZERO
                               COMPUTE KD1856 OF EF-AREA = KD1856 OF EF-AREA
                                                                + KE1124 OF EF-AREA(PI) * KE5001 OF EF-AREA(PI)
                               ON SIZE ERROR
                                   MOVE ALL "9" TO KD1856 OF EF-AREA
                           END-IF
                       END-IF
                   END-IF
               *****

```

W03: Homogeneity Vector

```

      IF ERR-INF = SPACE
      MOVE 1 TO ERR-INF
      MOVE 3019 TO MSG-NO
      END-IF
    END-COMPUTE
  END-IF
END-IF.
**** Disc. Total Amount L10990
  IF TKBN = CNS-EF
  IF KDU360 OF EF-AREA NOT = "1"
  IF KD1099 OF EF-AREA IS NUMERIC
  AND KE5001 OF EF-AREA(PI) IS NUMERIC
  AND KEC251 OF EF-AREA(PI) IS NUMERIC
  IF PI = 1
  MOVE ZERO TO KD1099 OF EF-AREA
  END-IF
  IF KE5001 OF EF-AREA(PI) NOT = ZERO
  COMPUTE KD1099 OF EF-AREA
    = KD1099 OF EF-AREA
    + KEC251 OF EF-AREA(PI)
  ON SIZE ERROR
  MOVE ALL "9" TO KD1099 OF EF-AREA
  IF ERR-INF = SPACE
  MOVE 1 TO ERR-INF
  MOVE 3019 TO MSG-NO
  END-IF
  END-COMPUTE
  END-IF
END-IF.
**** Transaction Total Amount L18540
  IF TKBN = CNS-EF
  IF KD1854 OF EF-AREA IS NUMERIC
  AND KD1857 OF EF-AREA IS NUMERIC
  AND KD1099 OF EF-AREA IS NUMERIC
  COMPUTE KD1854 OF EF-AREA = KD1857 OF EF-AREA
    - KD1099 OF EF-AREA
  ON SIZE ERROR
  MOVE ALL "9" TO KD1854 OF EF-AREA
  IF ERR-INF = SPACE
  MOVE 1 TO ERR-INF
  MOVE 3019 TO MSG-NO
  END-IF
  END-COMPUTE
  END-IF
END-IF.
**** Trns. Total Amount inclusive of Consumption Tax L18550
  IF TKBN = CNS-EF
  IF KD1855 OF EF-AREA IS NUMERIC
  AND KD1854 OF EF-AREA IS NUMERIC
  AND W10903 IS NUMERIC
  IF (KS0901 OF WBF-AREA = "1")
  IF KD1854 OF EF-AREA >= ZERO
  COMPUTE KD1855 OF EF-AREA
    = KD1854 OF EF-AREA * (1 + W10903) + 0.5
  ON SIZE ERROR
  MOVE ALL "9" TO KD1855 OF EF-AREA
  IF ERR-INF = SPACE
  MOVE 1 TO ERR-INF
  MOVE 3019 TO MSG-NO
  END-IF
  END-COMPUTE
  ELSE
  COMPUTE KD1855 OF EF-AREA
    = KD1854 OF EF-AREA * (1 + W10903) - 0.5
  ON SIZE ERROR
  MOVE ALL "9" TO KD1855 OF EF-AREA
  IF ERR-INF = SPACE

```

```

W04: Duplicate Vector
*-----*
*      Duplicate Vector Group 1 (Tail)      *
*-----*
*      Y17990  Wholesale Cost No.           *
*-----*
*      Y04360  Total Amount                 *
*-----*
Y04360-SEC SECTION.
Y04360-START.
  IF ERR-INF NOT = "1"
  AND GAMEN-NO OF W0204(PS2) = "KH201E"
  IF PPKBY-NO OF W0204(PS2) = 1
  OR PPKBY-NO OF W0204(PS2) = 12
  MOVE ZERO TO A04360 OF KH201E(PS1)
  ELSE
  IF KD1857 OF W03-HDIR - IS NUMERIC
  MOVE KD1857 OF W03-HDIR
  TO A04360 OF KH201E(PS1)
  END-IF
  END-IF
  END-IF.
Y04360-EXIT.
EXIT.
*-----*

```

```

W04: Homogeneity Vector
*-----*
*      L04360  Total Amount                 *
*-----*
L04360-SEC SECTION.
L04360-START.
  MOVE ZERO TO WK-KINGAKU.
  PERFORM VARYING PJ FROM 1 BY 1
  UNTIL PJ > 6
  IF A08311 OF KH201E(PS1 PJ) NOT = SPACE
  AND A09221 OF KH201E(PS1 PJ) IS NUMERIC
  AND A09221 OF KH201E(PS1 PJ) NOT = ZERO
  AND A10791 OF KH201E(PS1 PJ) IS NUMERIC
  AND A10791 OF KH201E(PS1 PJ) NOT = ZERO
  *
  AND A16721 OF KH201E(PS1 PJ) IS NUMERIC
  AND A16721 OF KH201E(PS1 PJ) NOT = ZERO
  COMPUTE WK-KINGAKU
  = WK-KINGAKU + A16721 OF KH201E(PS1 PJ)
  END-IF
  END-PERFORM.
  MOVE WK-KINGAKU TO A04360 OF KH201E(PS1).
L04360-EXIT.
EXIT.

```

Industrial Applicability

In the traditional software development the dependency on personal abilities is extremely high as well as having high personal ability is required for all staff involved, which has brought about fundamental problems. This invention has realized a development methodology, by theoretically grasping the consciousness action existing in the depths of the thinking method in relation to the cognitive action, and it is applicable to any fields including OS middle game, process control and business software.

Since the software structure which is induced by this invention is determined theoretically, it is recurrent and becomes the one and only. As a result, the software developed is not a black box any more, thereby eliminating human errors from the quality viewpoint, so that the system becomes stiff with the clear structure. Therefore, not only an exact estimate of software development is made possible development but also planning and the development management can be performed with ease.

Fig. 45 shows its effects. Compared with the traditional method, the development period is shortened by $1/2$ to $1/10$; the total development volume is compressed by 20% to 80%; the maintainability increases by 40 to 100 times; and the working efficiency is enhanced by 40 to 100 times.

In the traditional software, definitives comprised of screens, vouchers and files take up 30% and their logic takes up 70% of the whole software developed, thereby

necessitating a document respectively. However, in this invention, the homogeneity map which is corresponding to the above-mentioned logic in the traditional method does not require a document, so that it can cut the volume of documents by 70% compared with that of the traditional method.

Claims

1. The software production method comprising the following steps:

the step to determine a definitive identifier which is a screen necessary to the software to be produced;

the step to sort out words existing in said definitive identifier;

the step to create the homogeneity map in which all pallets necessary to the software have been plot in accordance with the process flow based on the definitive identifier;

the step to determine necessary files based on the sorted-out words and the created homogeneity map;

the step to create first tense control vectors which execute screen editing or file editing, second tense control vectors which determine the homogeneity route, and third tense control vectors which execute file updating for all the sorted-out words;

the step to create three kinds of the pallets which are made by binding the first, second and the third tense control vectors per screen unit, respectively, and to create pallet functions which execute each of the tense control vectors with autonomous significance inside each of the pallets; and

the step to assemble the three kinds of the pallet functions into a pallet chain function having the structure of transmitting a screen based on the pallet function concerned with the first tense control vector, receiving the

screen to execute the pallet function concerned with the second tense control vector, and determining, in accordance with this execution result, one homogeneity route from the plural homogeneity routes taking at least one homogeneity route for executing the pallet function concerned with the third tense control vector.

2. The processing apparatus comprising:

first means for possessing a first tense control vector to execute screen editing or file editing which is to be created for all words existing in the screen necessary to the software to be produced, a second tense control vector to determine the homogeneity route, and a third tense control vector to execute file updating;

second means for executing each of the tense control vectors with autonomous significance inside each of three kinds of pallets which are made by binding the first, the second and the third tense control vectors per screen unit, respectively; and

third means for transmitting a screen based on the execution of the second means concerned with the first tense control vector, receiving the screen to execute the second means concerned with the second tense control vector, and determining, in accordance with this execution result, one homogeneity route from the plural homogeneity routes taking at least one homogeneity route for executing the second means concerned with the third tense control vector.

3. The recording medium on which a program is recorded, said program comprising;

first means for possessing a first tense control vector to execute screen editing or file editing which is to be created for all words existing in the screen necessary to the software to be produced, a second tense control vector to determine the homogeneity route, and a third tense control vector to execute file updateing;

second means for executing each of the tense control vectors with autonomous significance inside each of three kinds of pallets which are made by binding the first, the second and the third tense control vectors per screen unit, respectively; and

third means for transmitting a screen based on the execution of the second means concerned with the first tense control vector, receiving the screen to execute the second means concerned with the second tense control vector, and determining, in accordance with this execution result, one homogeneity route from the plural homogeneity routes taking at least one homogeneity route for executing the second means concerned with the third tense control vector.

4. The process method comprising the following steps:

the step to determine a screen to be displayed;

the step to activate a first tense control vector (W04 pallet) for implementing screen editing or file editing;

the step to display the screen edited based on the first tense control vector (W04 pallet);

the step to receive the operational contents on the screen displayed;

the step to activate a second tense control vector (W02

pallet) for determining, the homogeneity route based on the operational contents;

the step to judge the homogeneity route in accordance with the homogeneity route determined by the second tense control vector (W02 pallet); and

the step to activate a third tense control vector (W03 pallet) for implementing file updating based on the judging results of the homogeneity route.

5. The processing apparatus comprising:

the means for determining a screen to be displayed;

the means for activating a first tense control vector (W04 pallet) for implementing screen editing or file editing;

the means for displaying the screen edited based on the first tense control vector (W04 pallet);

the means for receiving the operational contents on the screen displayed;

the means for activating a second tense control vector (W02 pallet) for determining the homogeneity route based on the operational contents;

the means for judging the homogeneity route in accordance with the homogeneity route determined by the second tense control vector (W02 pallet); and

the means for activating a third tense control vector (W03 pallet) for implementing file updating based on the judging results of the homogeneity route.

6. The recording medium on which a program is recorded, said program comprising:

the means for determining a screen to be displayed;

the means for activating a first tense control vector (W04 pallet) for implementing screen editing or file editing;

the means for displaying the screen edited based on the first tense control vector (W04 pallet);

the means for receiving the operational contents on the screen displayed;

the means for activating a second tense control vector (W02 pallet) for determining the homogeneity route based on the operational contents;

the means for judging the homogeneity route in accordance with the homogeneity route determined by the second tense control vector (W02 pallet); and

the means for activating a third tense control vector (W03 pallet) for implementing file updating based on the judging results of the homogeneity route.

Abstract

Firstly, definitive identifiers are determined, which are the screen needed for the software to be produced. Next, words existing in the definitive identifier are sorted out, as well as the homogeneity map is created where all pallets needed for the software are placed according to the process flow based on the definitive identifier. Next, the necessary file is determined based on the sorted-out words and the created homogeneity map. Next, the following tense control vectors are created: the first tense control vector that implements screen editing or file editing for all the sorted-out words; the second tense control vector that determines the homogeneity route for all the sorted-out words; and the third tense control vector that implements file updating. Next, the three kinds of pallets are created, which are made by binding the first, the second and the third tense control vectors per screen unit, respectively, as well as the pallet function is created, which executes each of the tense control vectors inside each pallet with autonomous significance. And, the three kinds of the pallet functions are assembled into the pallet chain function with the structure of transmitting a screen based on the pallet function concerned with the first tense control vector, receiving the screen to execute the pallet function concerned with the second tense control vector, and determining, in accordance with this execution result, one homogeneity route from the plural homogeneity routes taking at least one homogeneity route for executing the pallet

function concerned with the third tense control vector. Hereupon, with this software, it is not necessary to create tense control vectors for all words, and if tense control vectors are created for a certain number of words, it operates within such scope.

FIG. 1

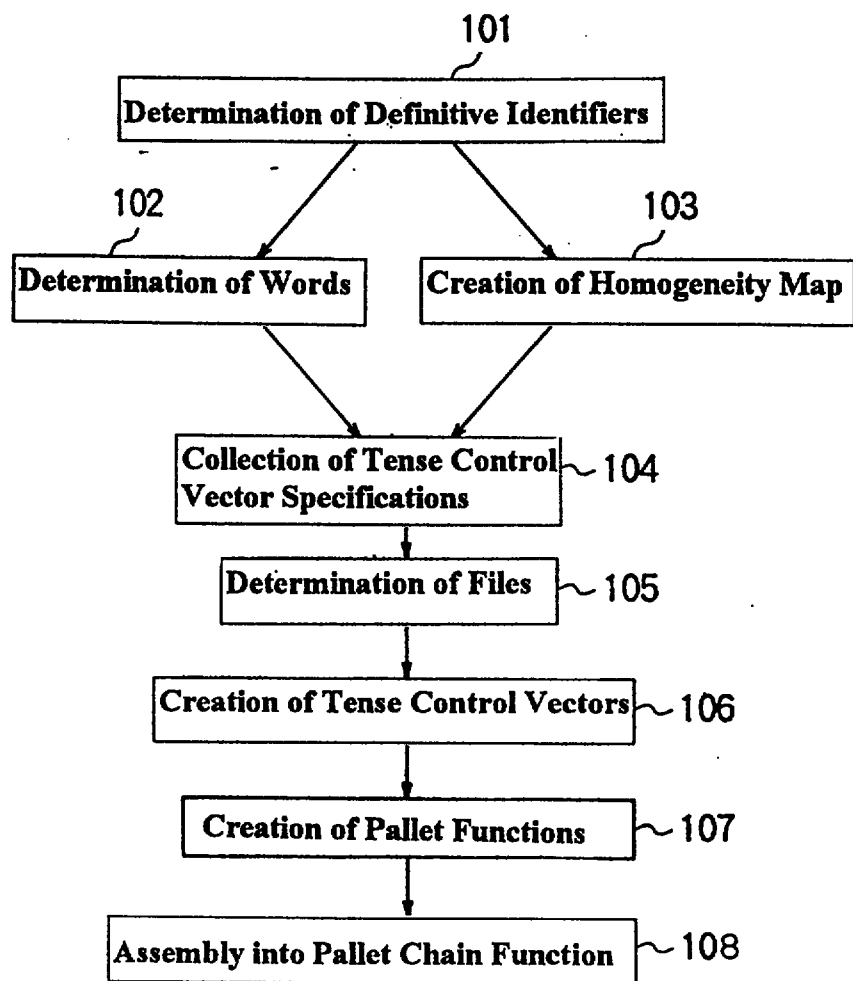


FIG. 2

[illegible]

FIG. 3

[illegible]

FIG. 4

[illegible]

FIG. 5

[illegible]

FIG. 6

1	2	3	4	5	6	7	8
1234567890123456789012345678901234567890123456789012345678901234567890							
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							
16							
17							
18							
19							
20							
21							
22							
23							
24							
1	2	3	4	5	6	7	8
1234567890123456789012345678901234567890123456789012345678901234567890							

FIG. 7

[illegible]

FIG. 8

行	桁	130-140										141-150										151-160										161-170										161-170										180-190						
		3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7												
1	1																																																									
	2																																																									
	3																																																									
	4																																																									
	5																																																									
	6																																																									
2	7																																																									
	8																																																									
	9	Your Order No.																				Inst.										City										Shipment										Sales						
	10	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X											N										X								X										
3	11	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X											N										X								X											
	12	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X											N										X								X											
	13																																																									
	14																																																									
	15																																																									
	16																																																									
	17	Your Order No.																				Inst.										City																										
	18	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X											N										N																			
	19	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X											N										N																				
	20	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X											N										N																				

FIG. 9

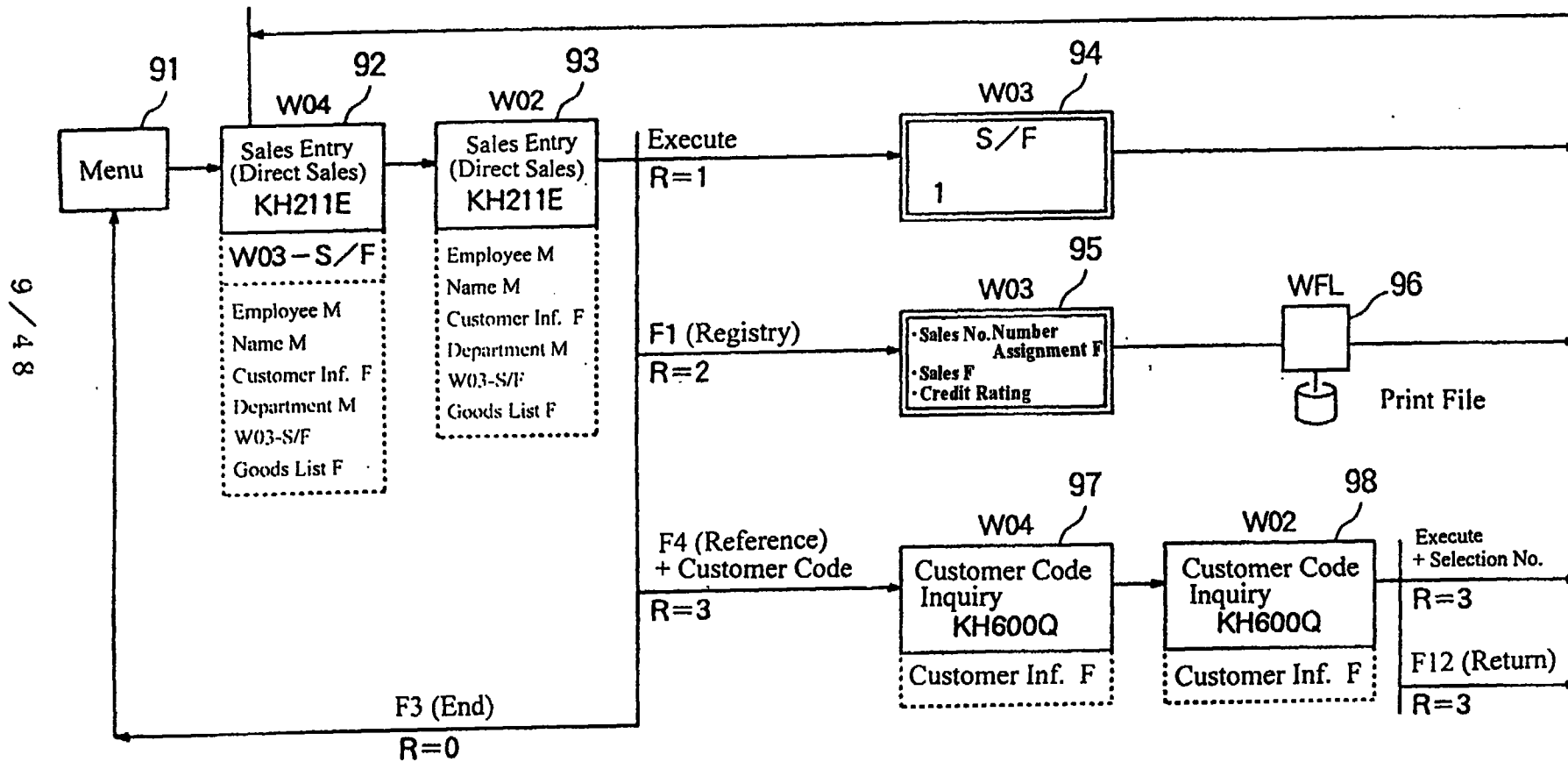


FIG. 10

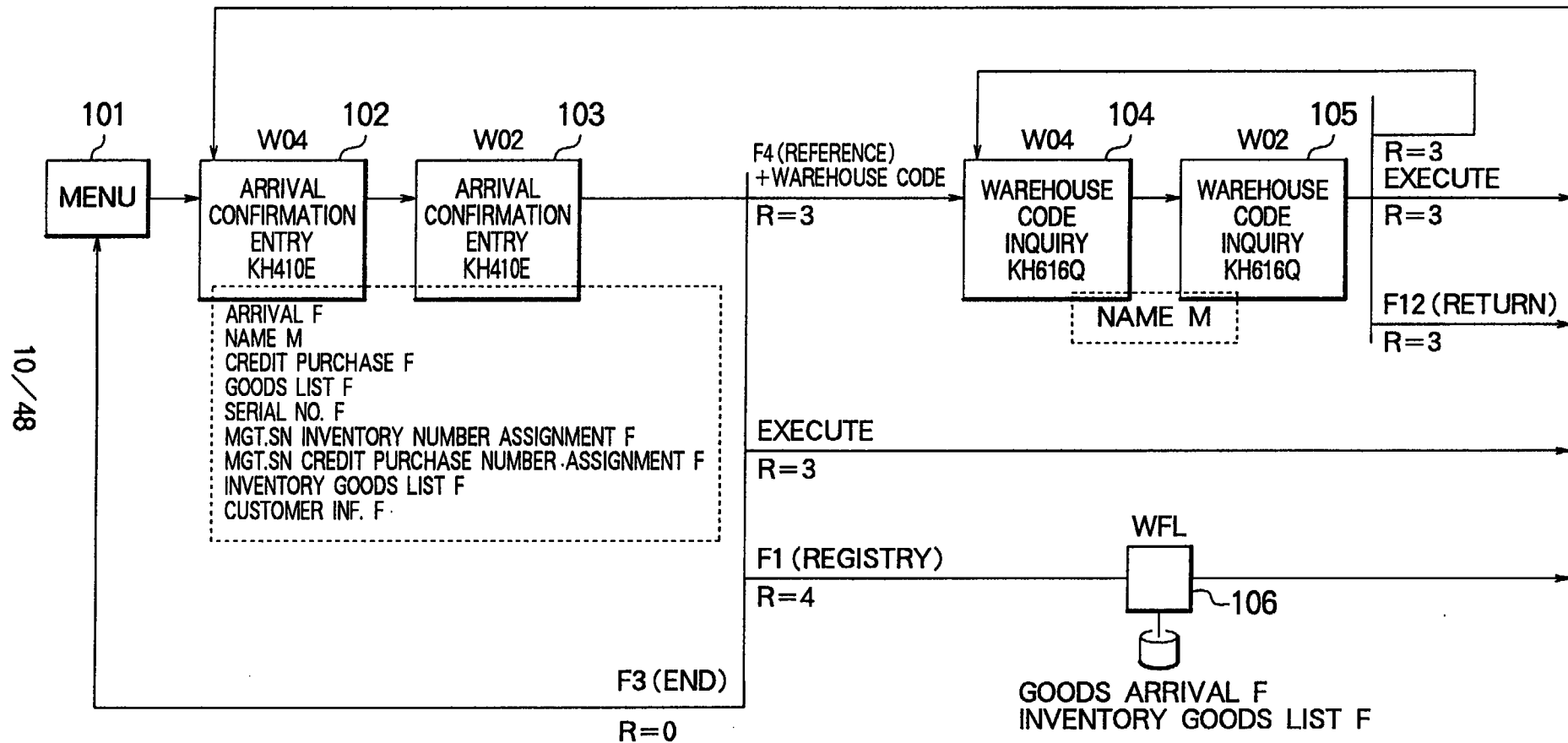


FIG. 11

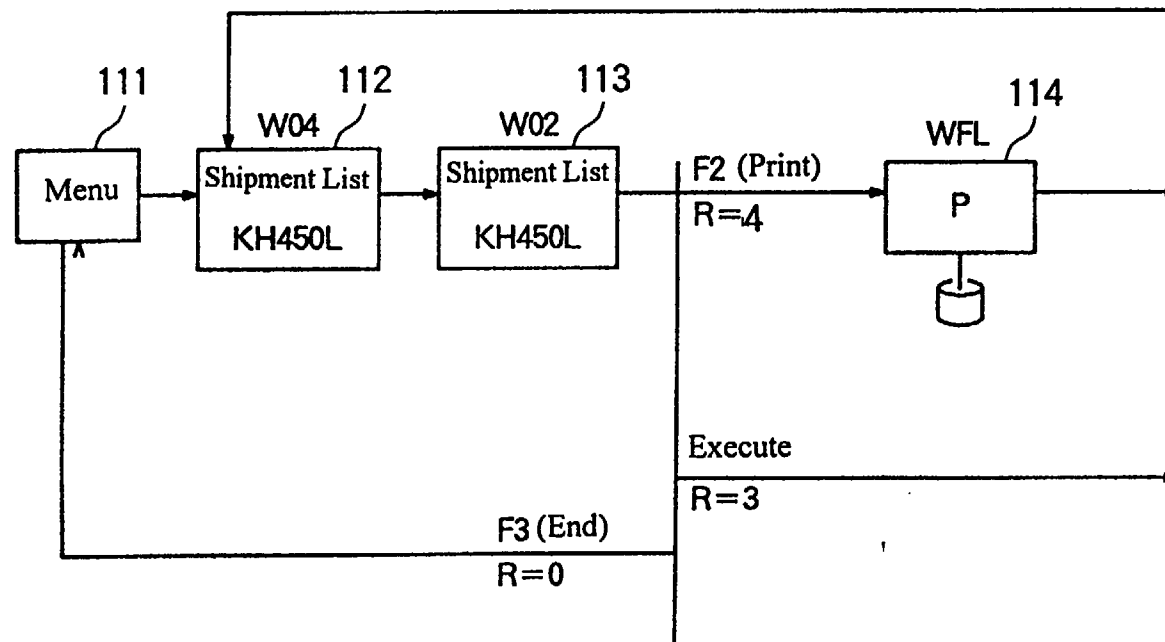


FIG. 12

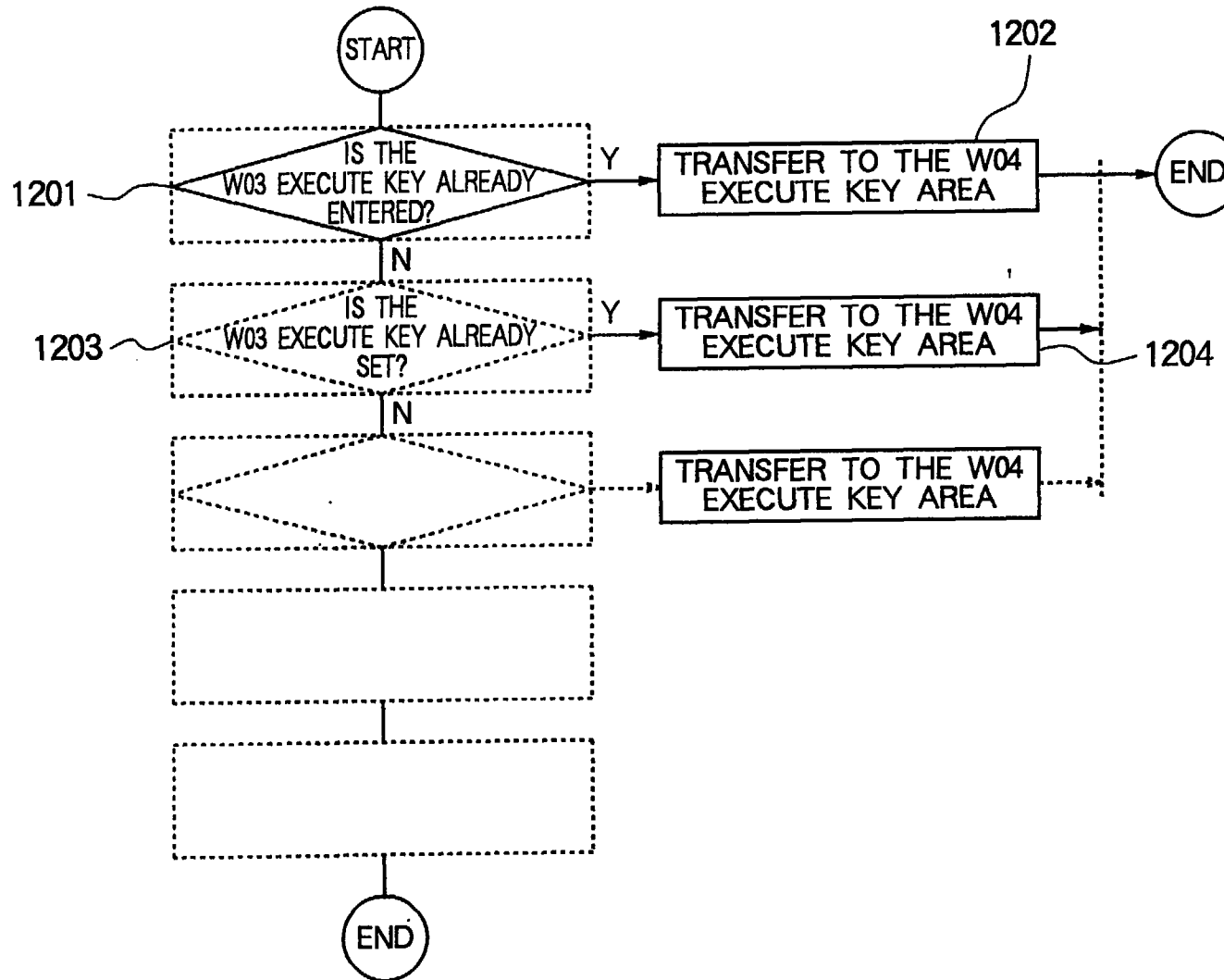


FIG. 13

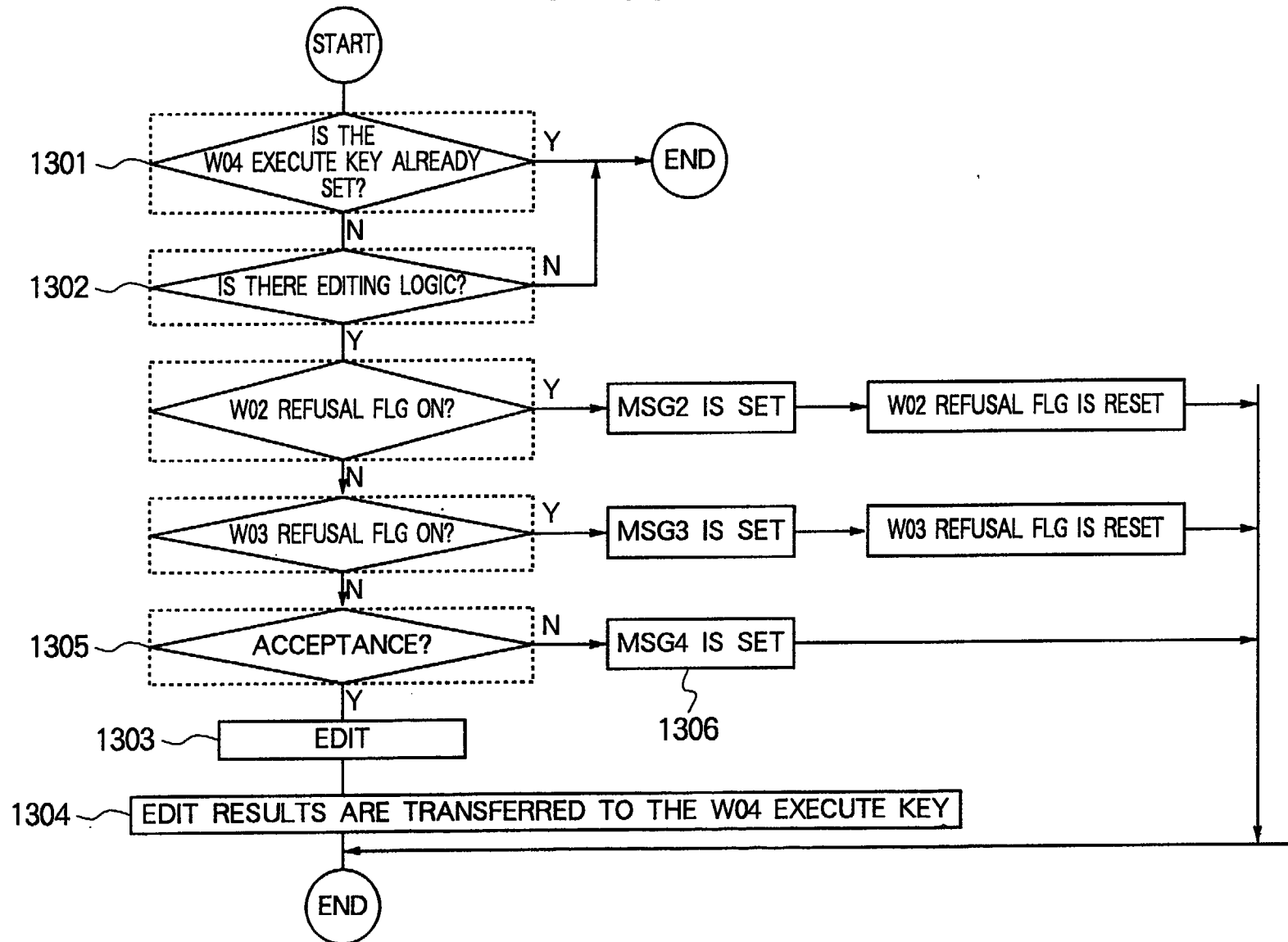


FIG. 14

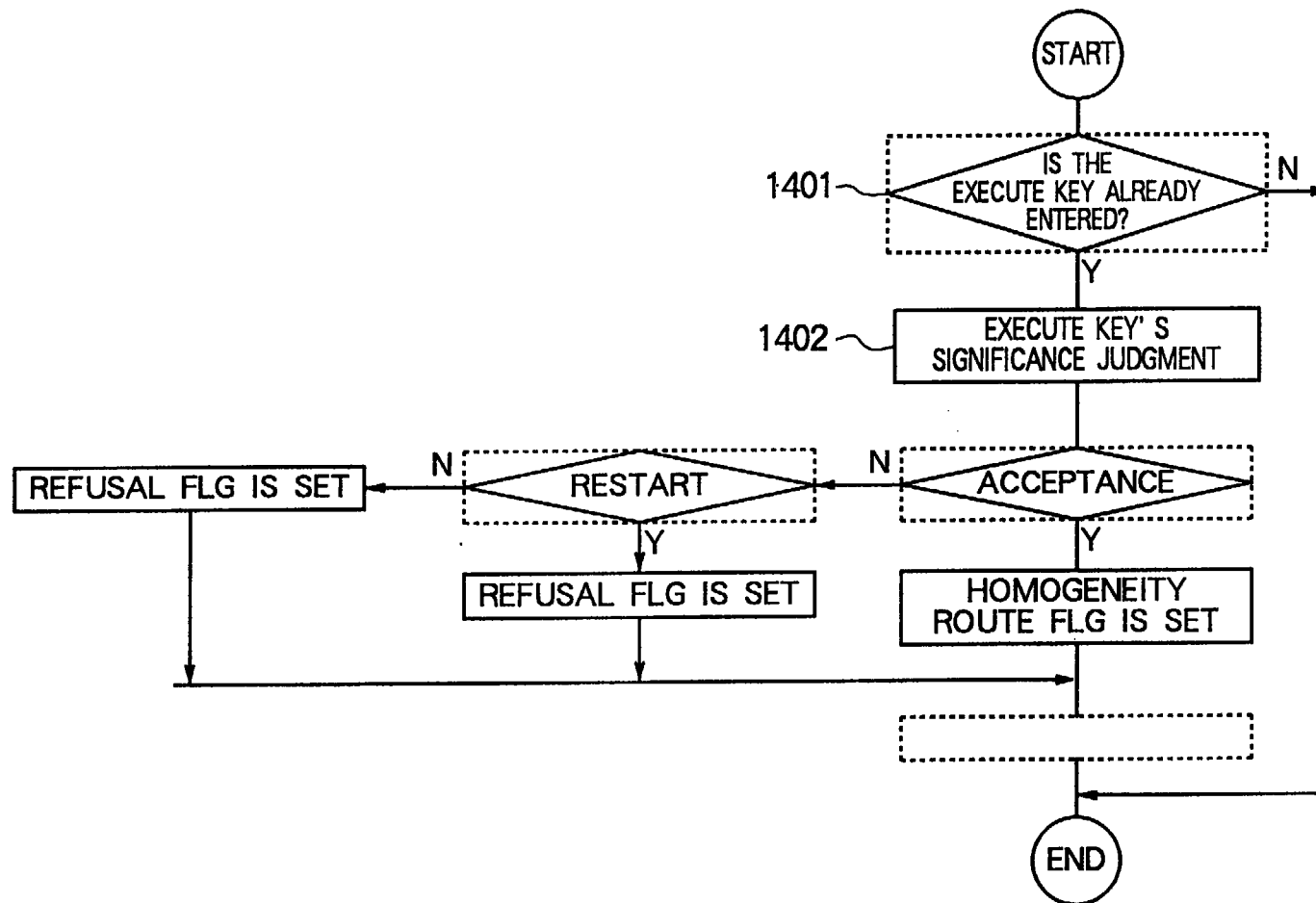


FIG. 15

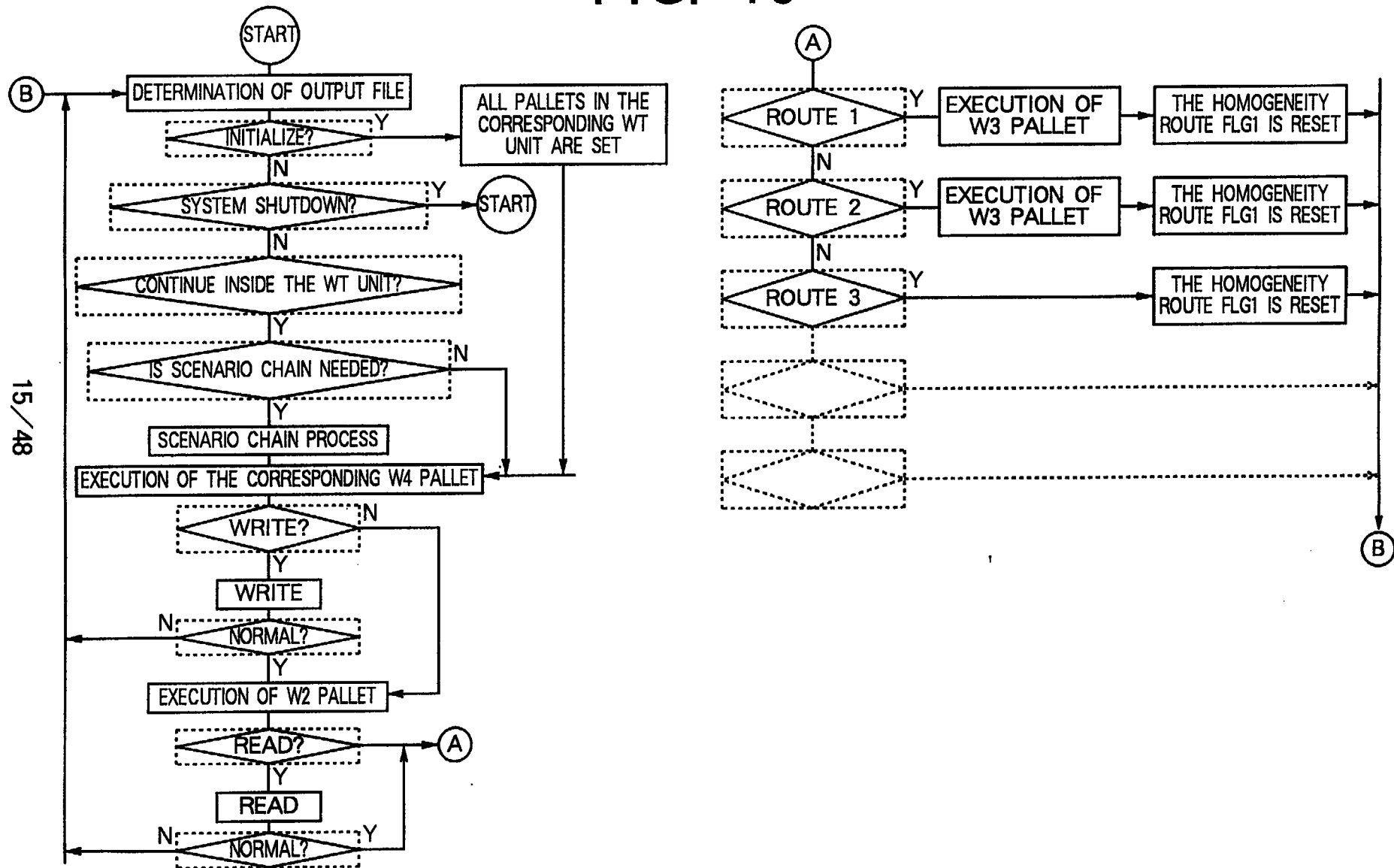


FIG. 16

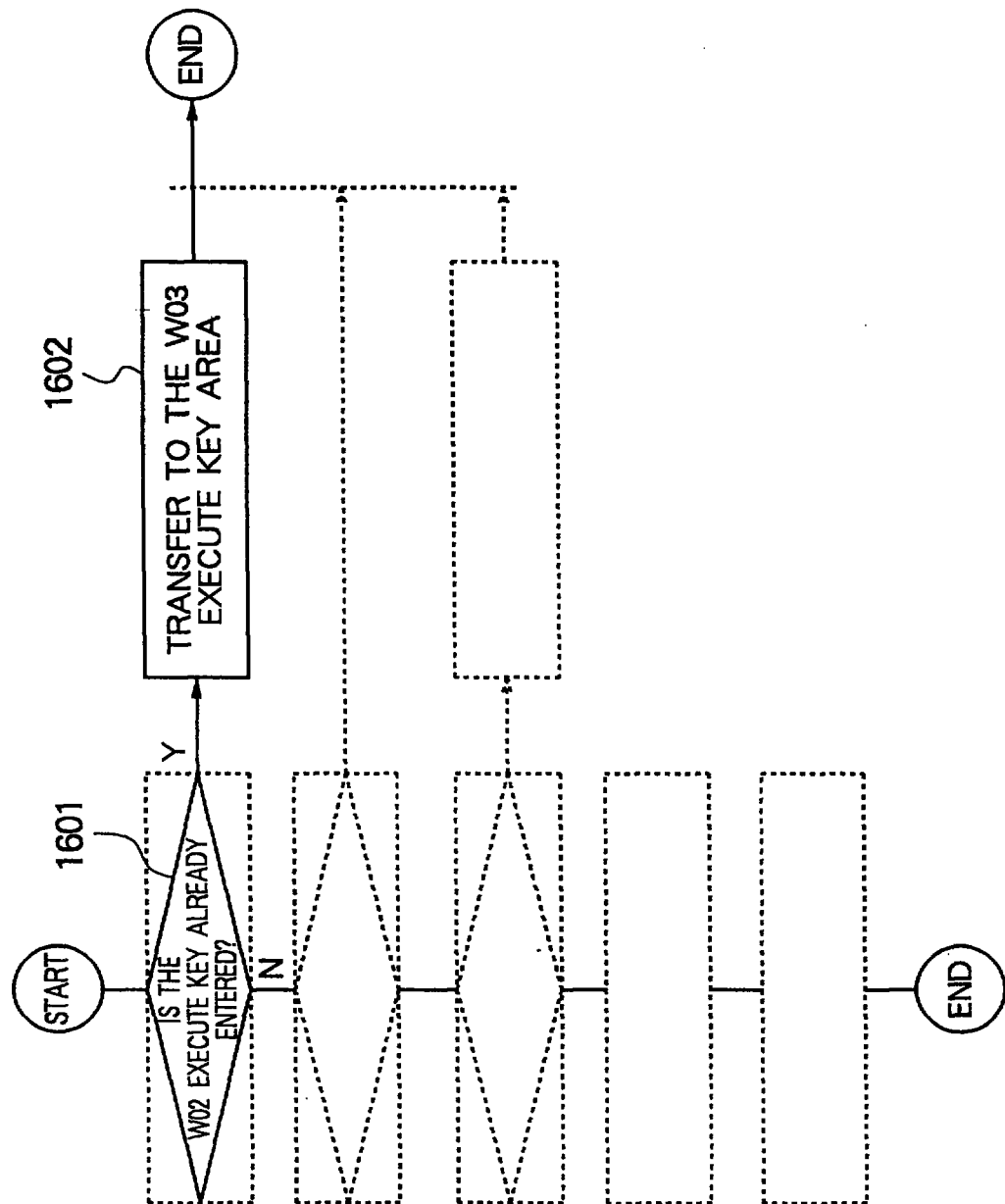


FIG. 17

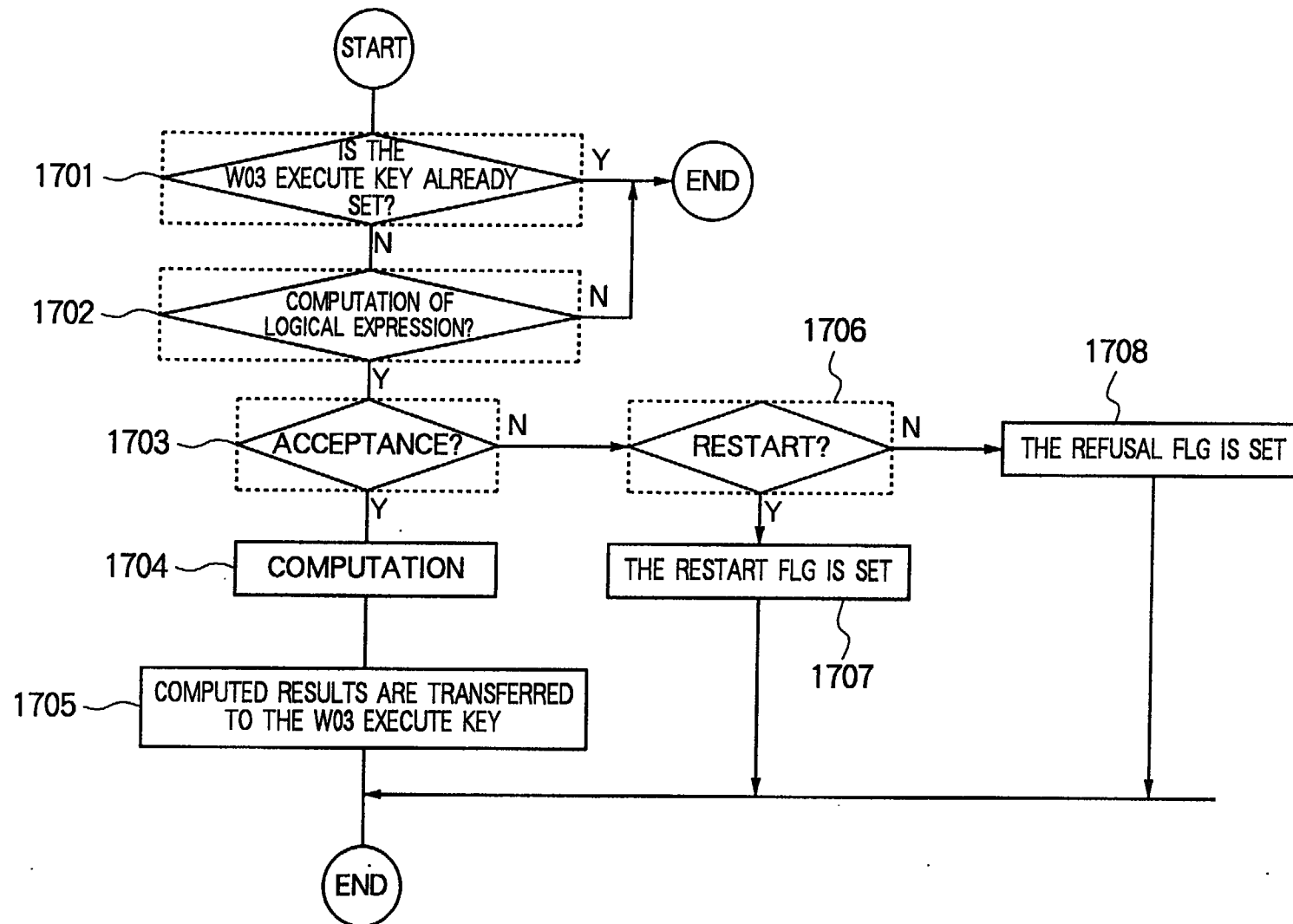


FIG. 18

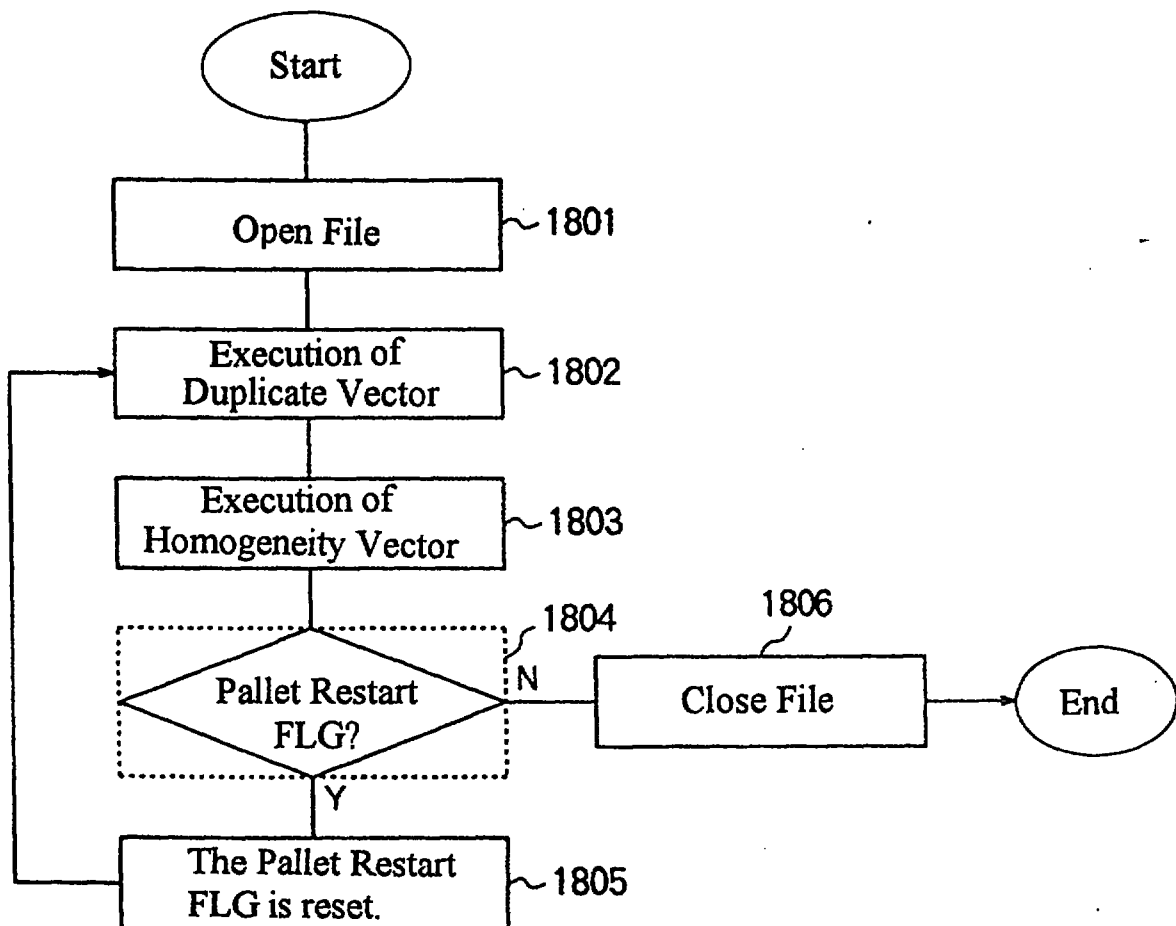


FIG. 19

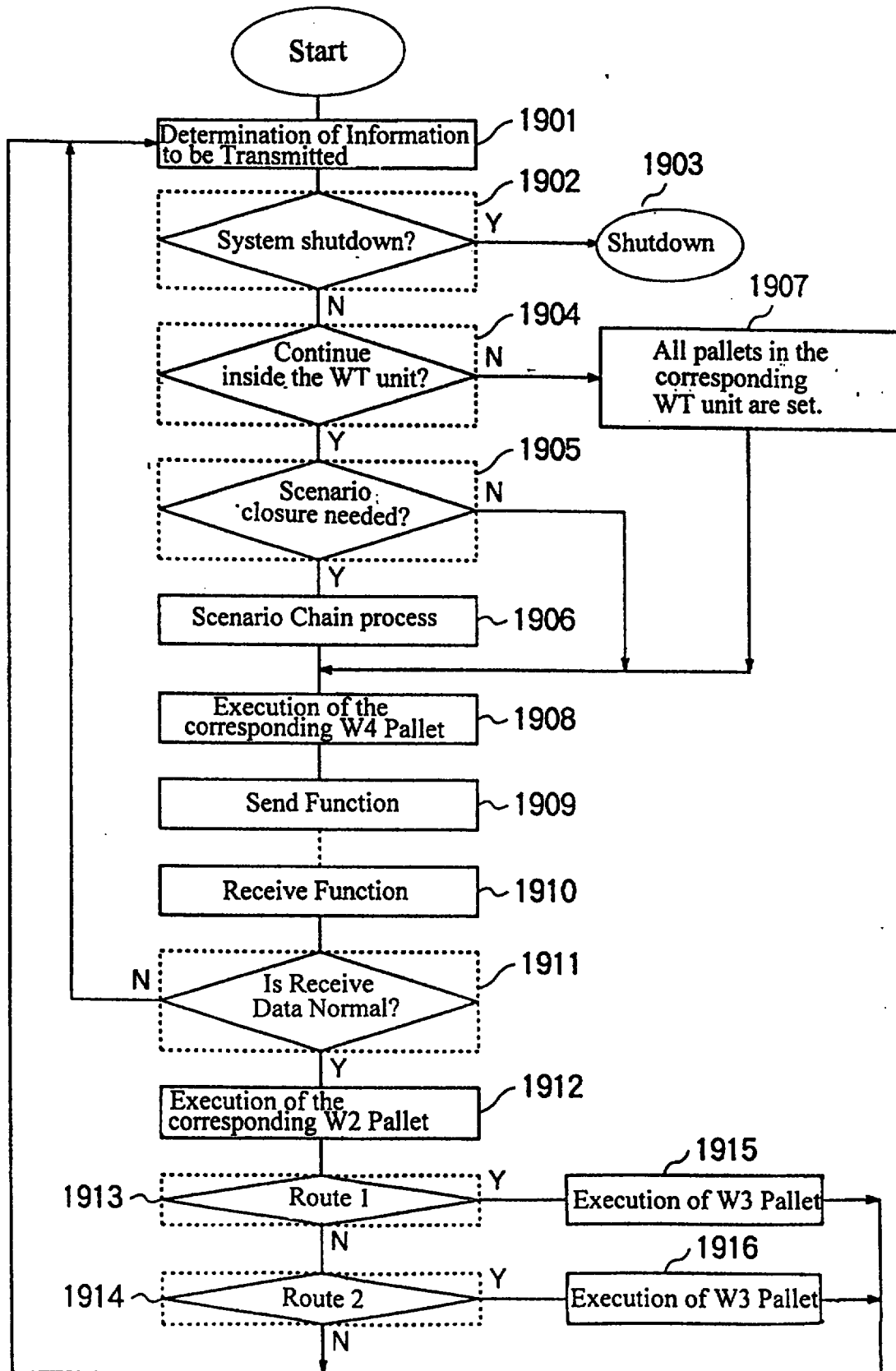


FIG. 20

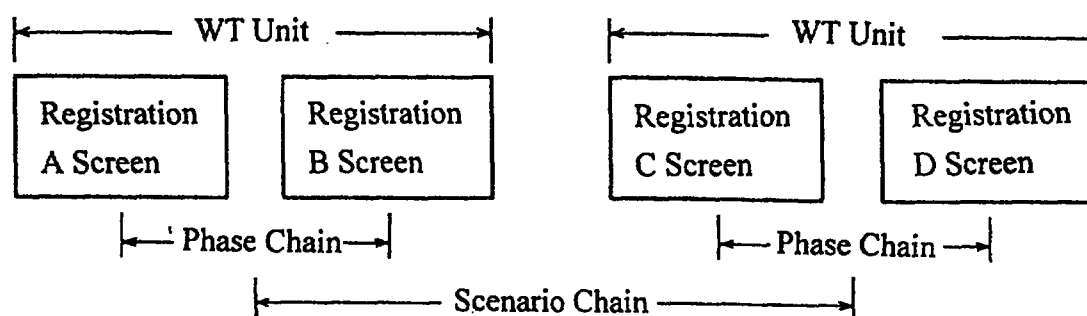


FIG. 21

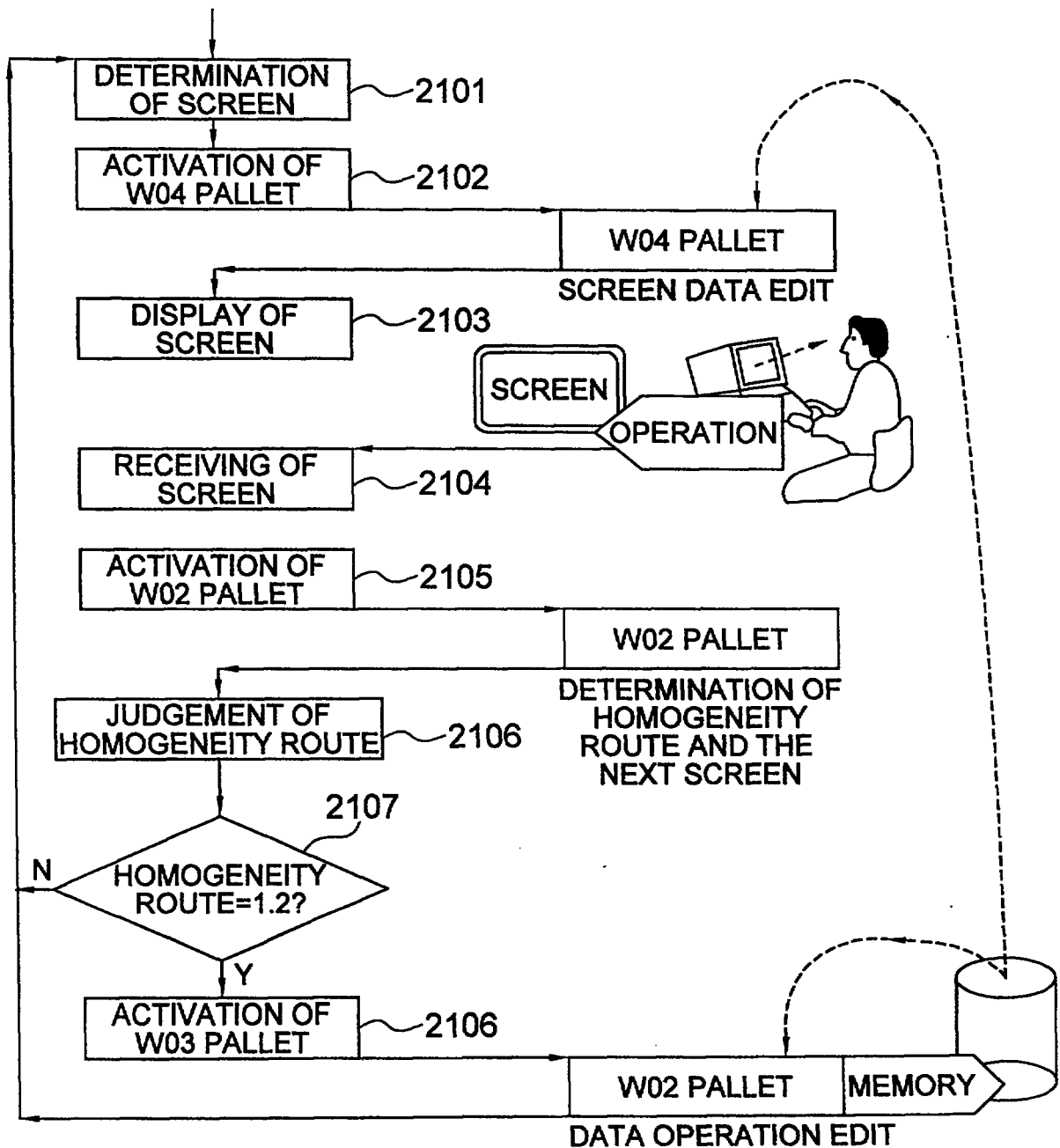


FIG. 22

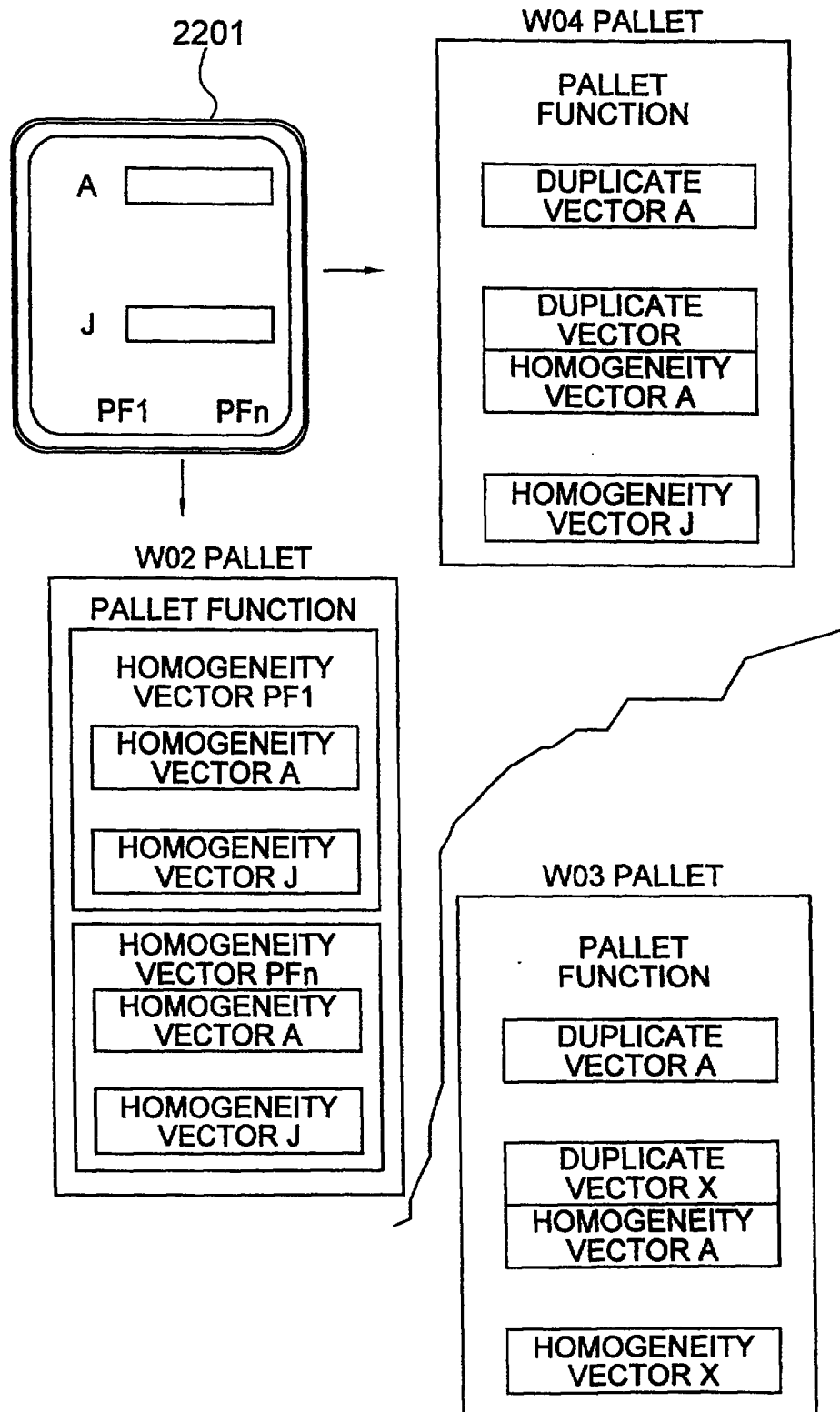


FIG. 23

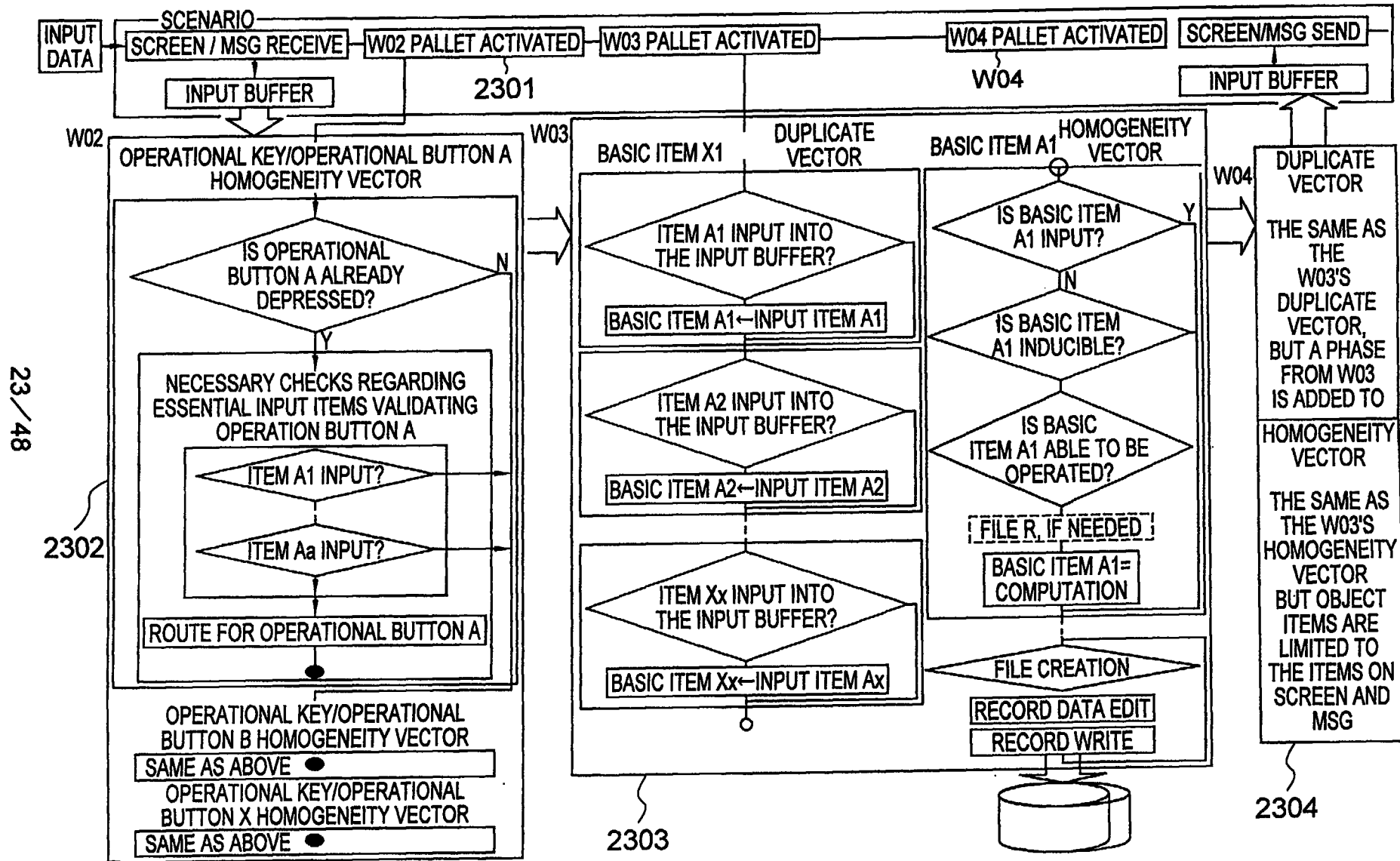


FIG. 24

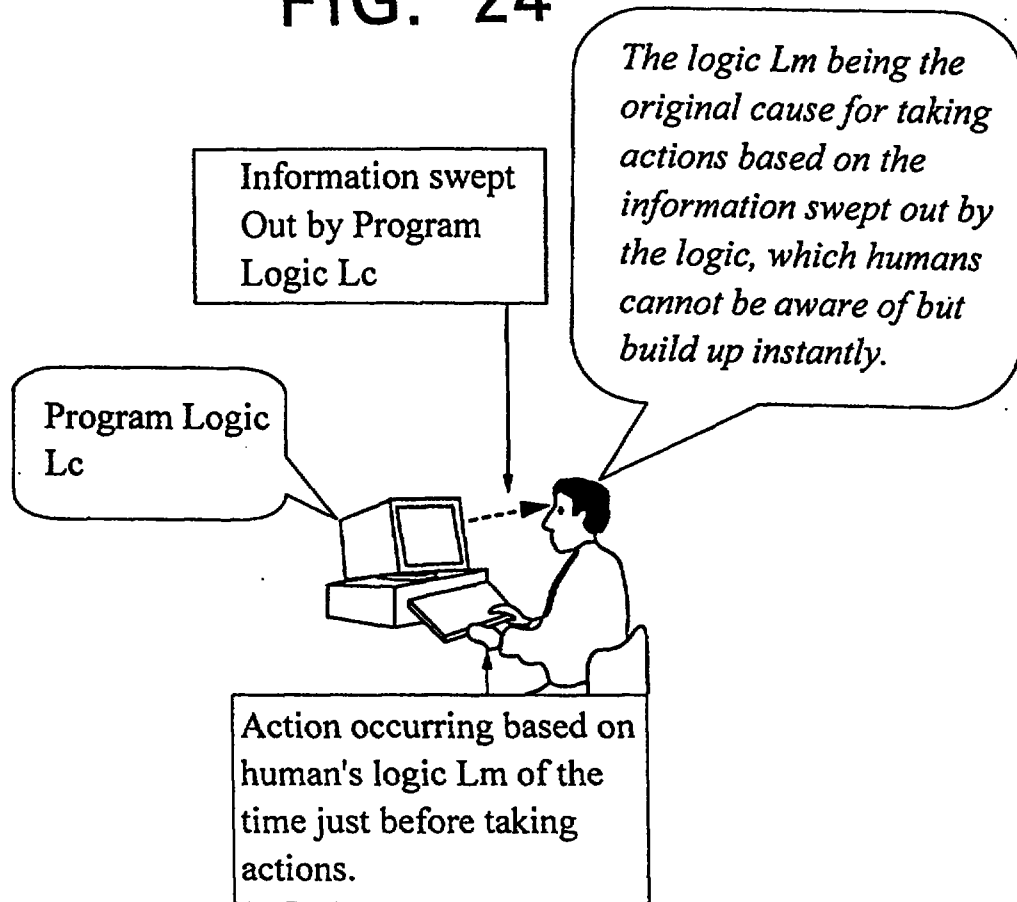


FIG. 25

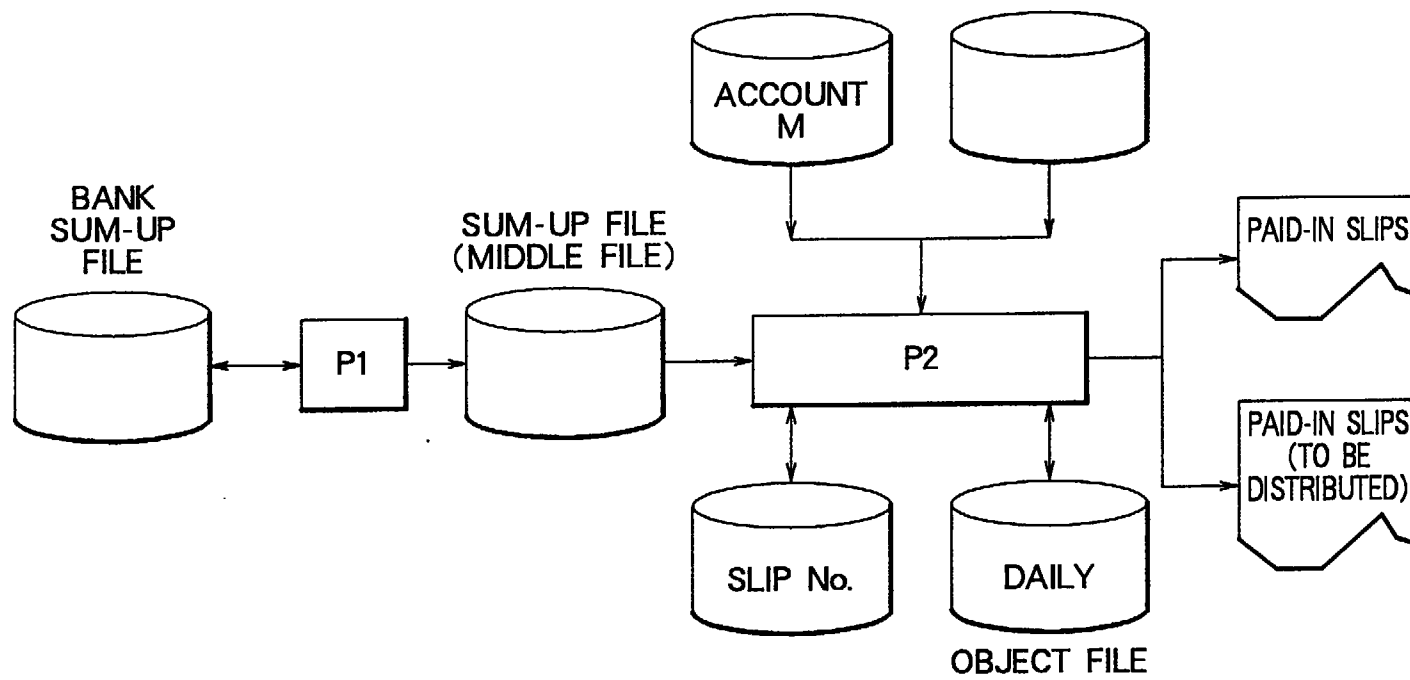


FIG. 26

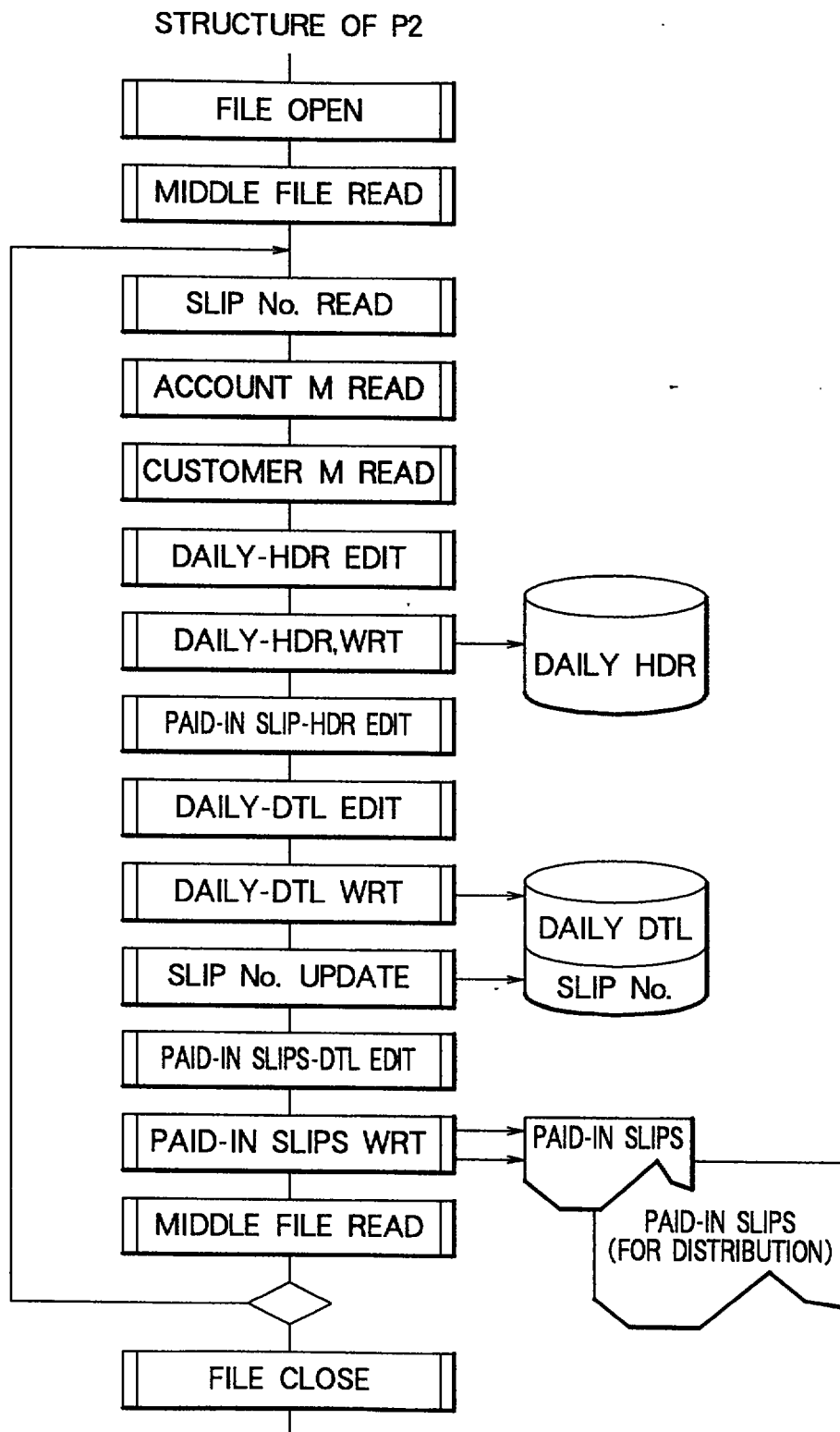
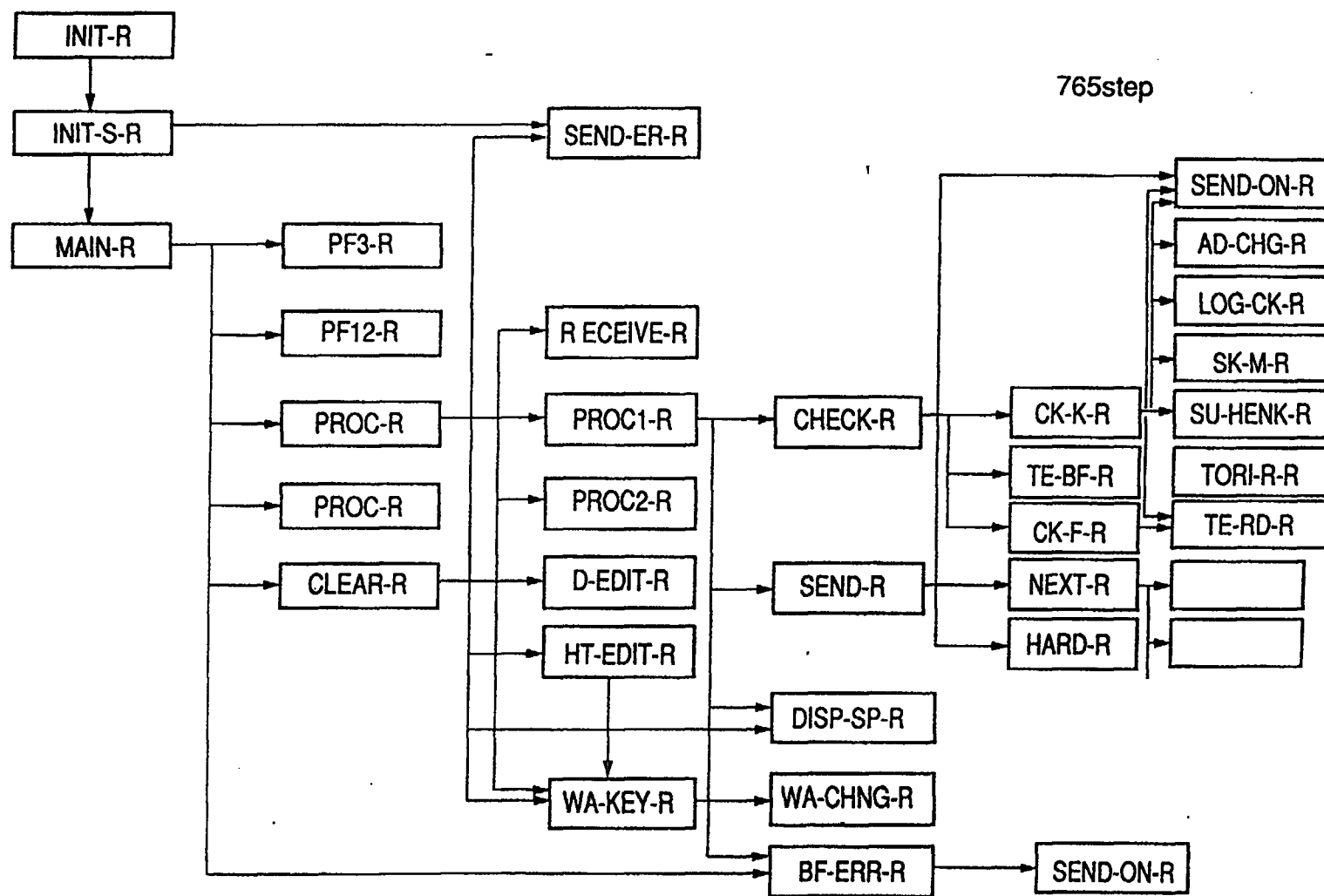


FIG. 27



765step

FIG. 28

Bill Inquiry Screen

A

B

: : ; ; X

[PF1] [PF3] [PF4] [PF12]

650step

W02

150step

PF1	A	B	:	:
			:	X

PF3	A	B	:	:
			:	X

PF4	A	B	:	:
			:	X

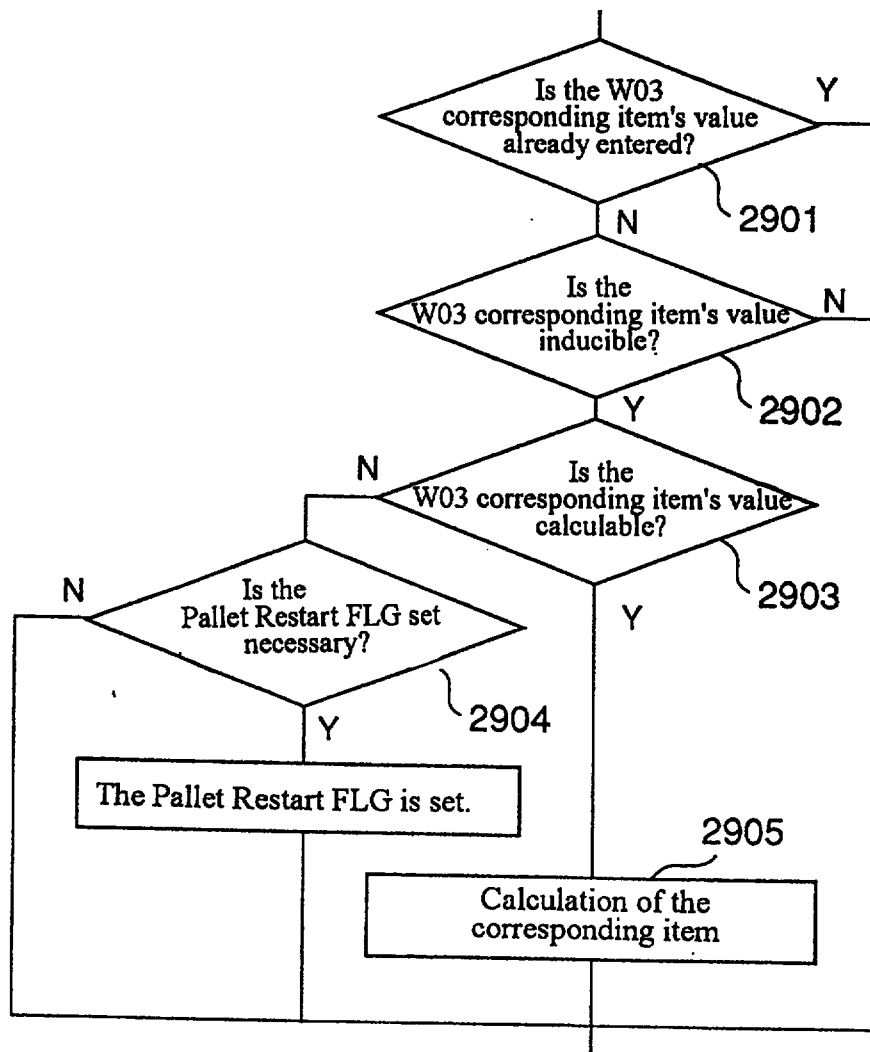
PF12	A	B	:	:
			:	X

W04

500step

a
:
x
A
:
X

FIG. 29



The flowchart illustrates the data processing system, showing the flow of data and control signals between various components. The components are organized into four main functional blocks, each containing a window (W02, W03, W04) and a data input section.

- Block 1 (Top Left):** Contains a **SCREEN** and a **W02** window. The **SCREEN** is connected to the **W02** window via a **DATA INPUT** line. The **W02** window contains **INPUT DATA IN-TAKE** and **DETERMINATION OF HOMOGENEITY ROUTE**.
- Block 2 (Top Middle):** Contains a **W03** window. The **W03** window contains **W02 DATA IN-TAKE** and **COMPUTATION WITH THE N-TAKE DATA**.
- Block 3 (Top Right):** Contains a **W04** window. The **W04** window contains **W02/03 DATA IN-TAKE** and **EDIT WITH THE IN-TAKE DATA**.
- Block 4 (Bottom):** Contains a **W04** window and a **REFERENCE SCREEN**. The **W04** window contains **W02 DATA IN-TAKE** and **EDIT WITH THE IN-TAKE DATA**. The **REFERENCE SCREEN** is connected to the **W04** window via a **DATA INPUT** line.

The flow of data and control signals is as follows:

- Registration (R=2):** A solid arrow labeled **REGISTRATION R=2** points from the **DETERMINATION OF HOMOGENEITY ROUTE** in Block 1 to the **W02 DATA IN-TAKE** in Block 2.
- Reference (R=3):** A solid arrow labeled **REFERENCE R=3** points from the **DETERMINATION OF HOMOGENEITY ROUTE** in Block 1 to the **W02 DATA IN-TAKE** in Block 4.
- Execution (R=3):** A solid arrow labeled **EXECUTE R=3** points from the **DETERMINATION OF HOMOGENEITY ROUTE** in Block 4 to the **W02 DATA IN-TAKE** in Block 3.
- Feedback Loops:** Dashed arrows indicate feedback paths from the **COMPUTATION WITH THE N-TAKE DATA** in Block 2 back to the **W02 DATA IN-TAKE** in Block 1 and Block 3, and from the **EDIT WITH THE IN-TAKE DATA** in Block 3 back to the **W02 DATA IN-TAKE** in Block 2.

FIG. 31

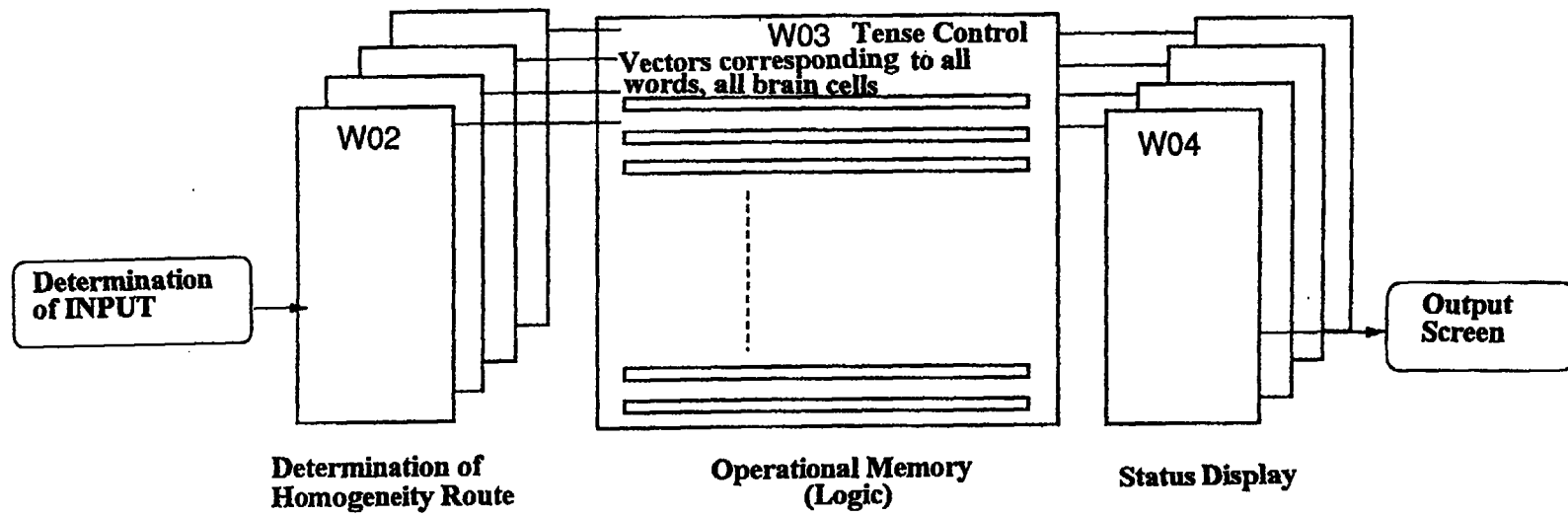


FIG. 32

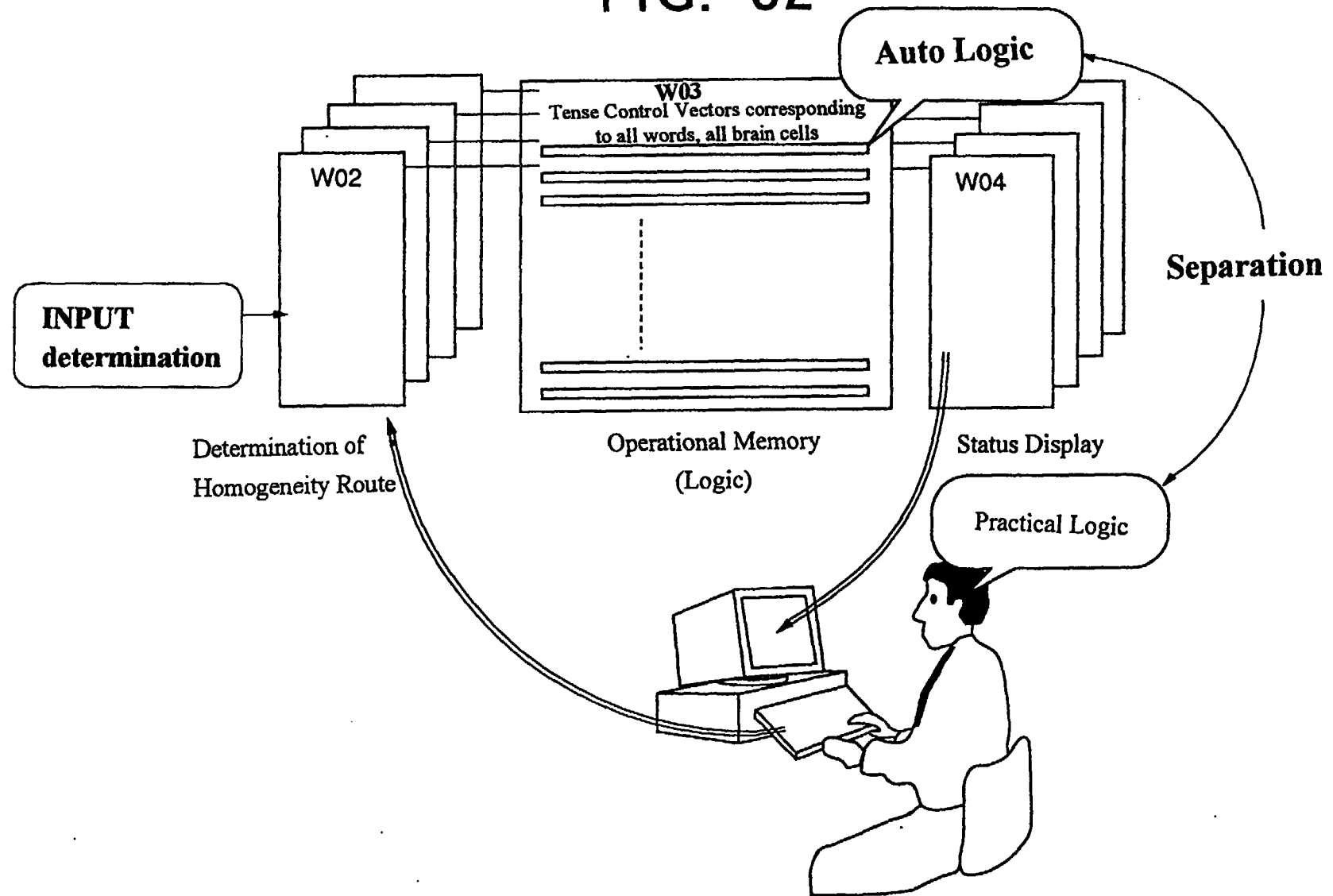


FIG. 33

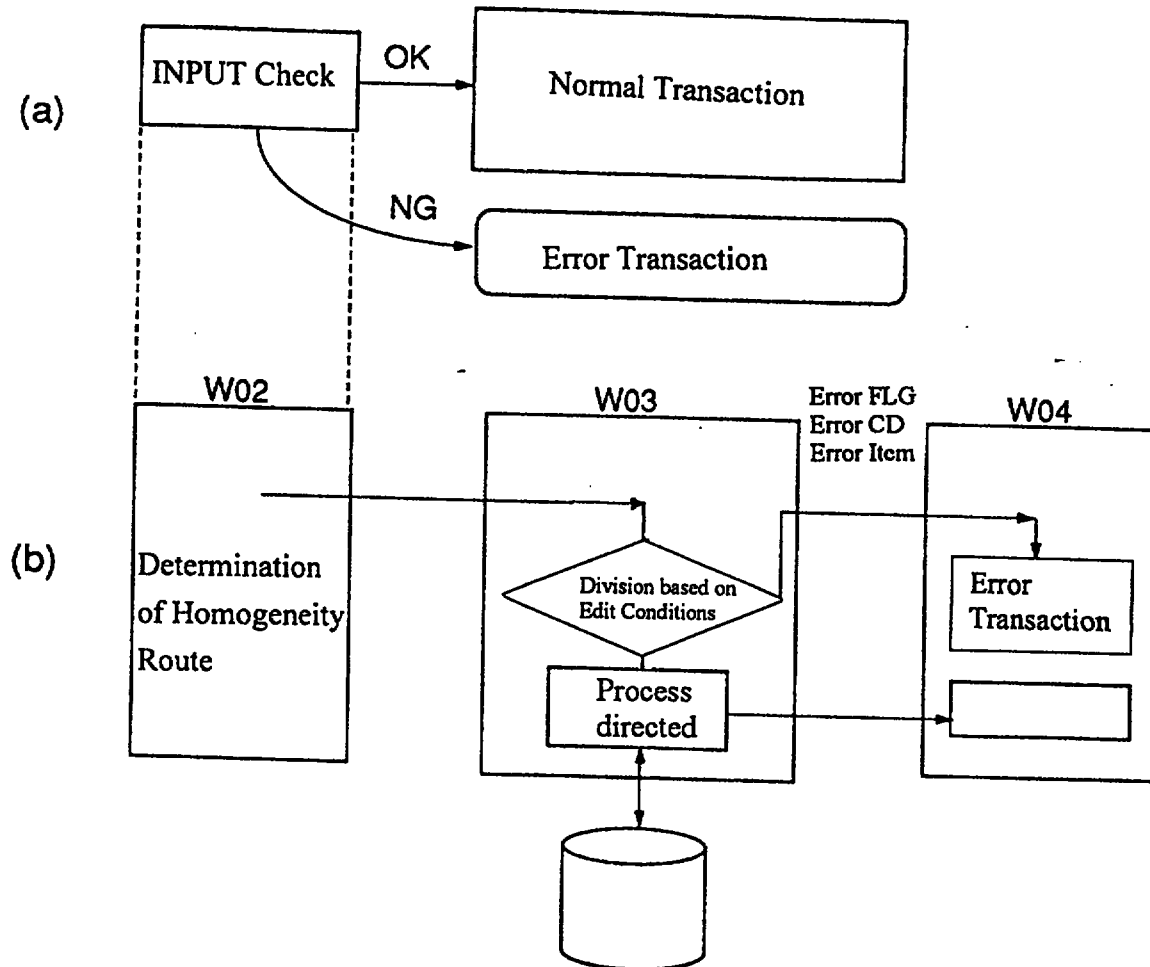


FIG. 34

(a)

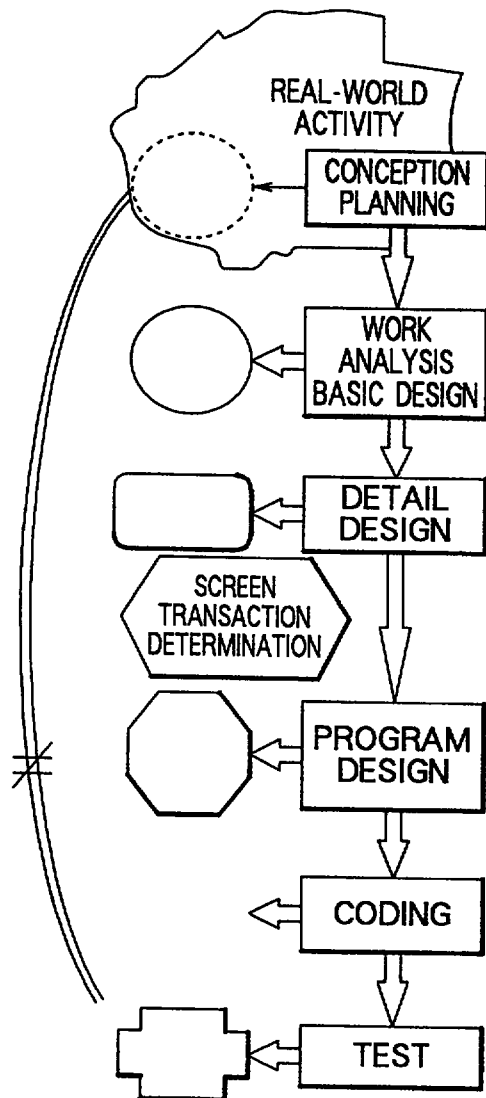


FIG. 34

(b)

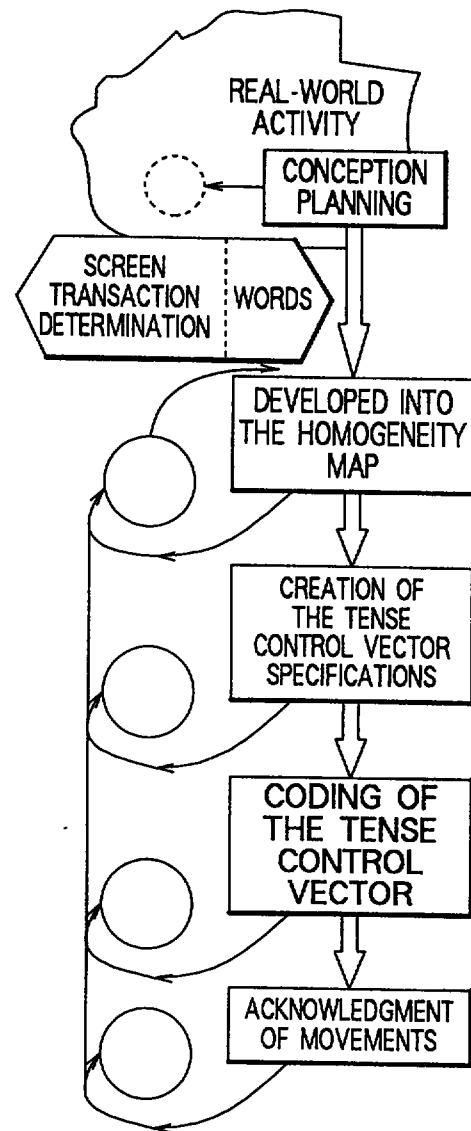


FIG. 35

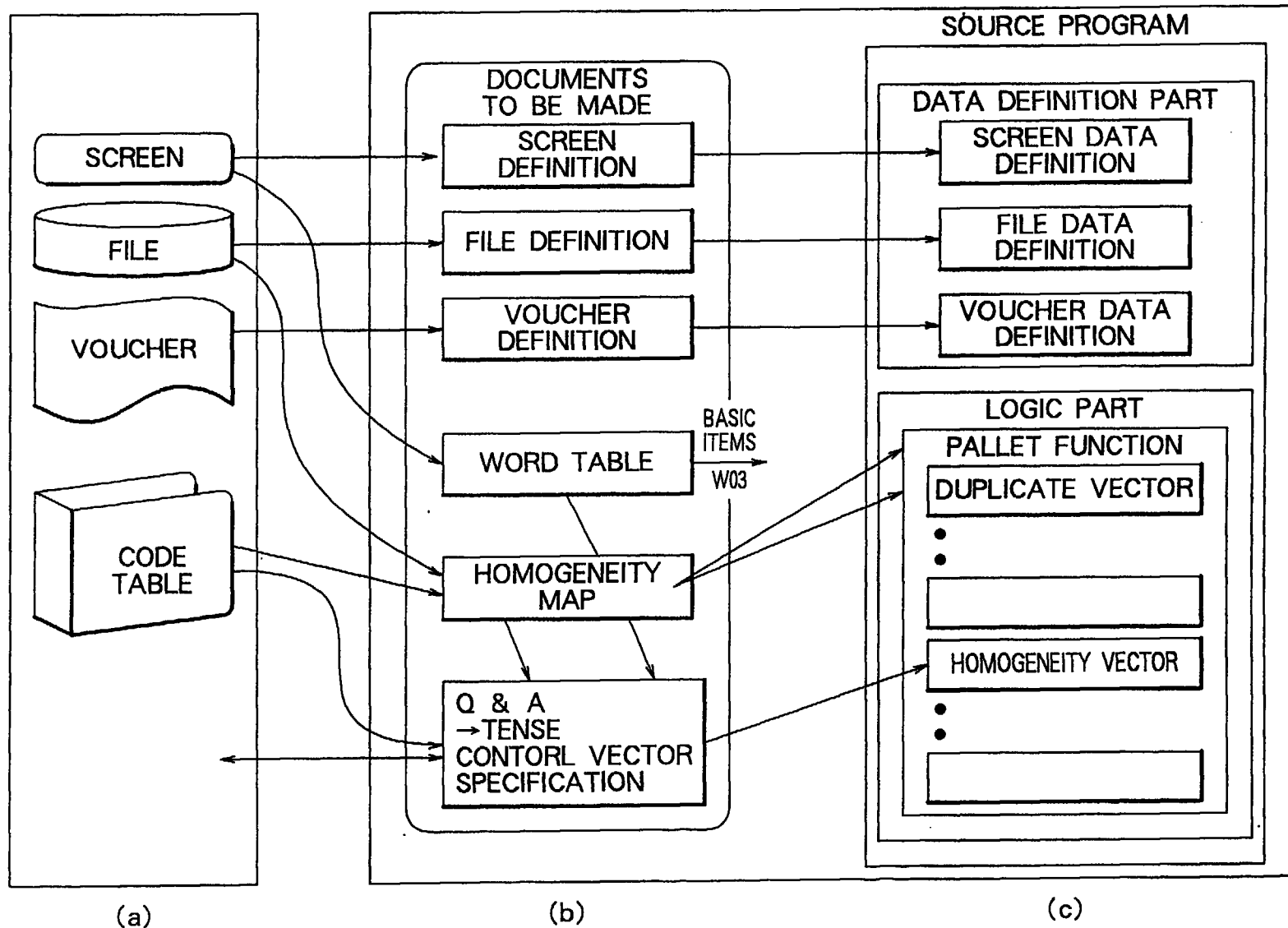


FIG. 36

36/48

FIG. 37

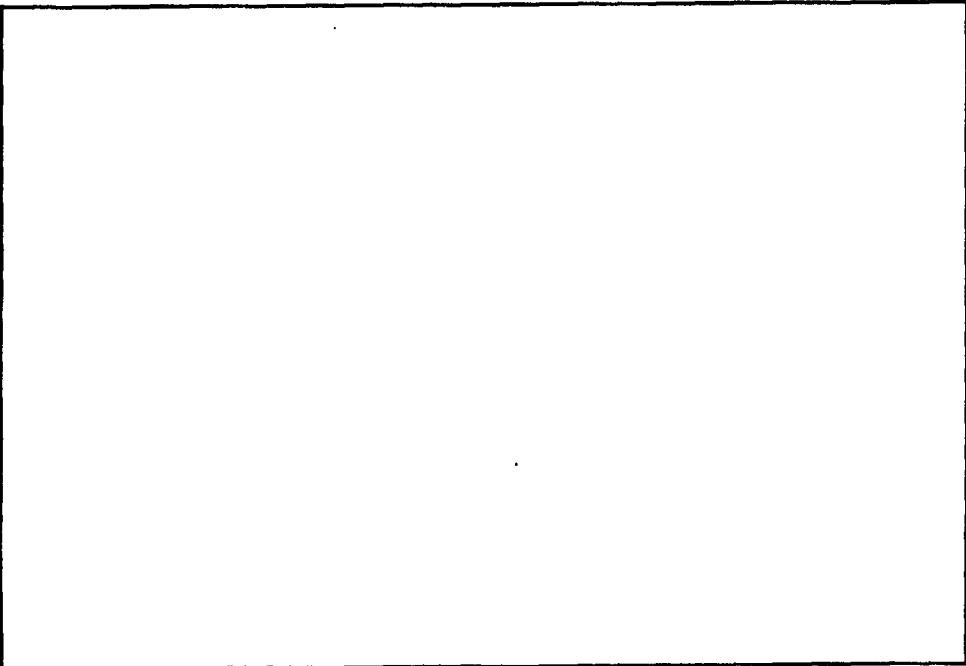
VOUCHER NAME:		IDENTIFIER:		OUTPUT CONTROL METHOD [NEW LINE , NEW PAGE ETC.]	
WORD PLACEMENT DIAGRAM (LAYOUT)					
010203040506 61626364656667					
01					
02					
03					
04					
.....					
.....					
.....					
.....					
.....					
.....					
.....					
.....					
.....					
.....					
.....					
57					
58					
59					
60					
VOUCHER OUTPUT CONDITIONS	OUTPUT POINT OUTPUT LOCATION OUTPUT DEVICE KINDS OF FORM OUTPUT TIME				REMARKS

FIG. 38

FILE NAME		IDENTIFIER	FILE CLASSIFICATION	$\alpha/\beta/\gamma$	BELONGING AREA	W02/W03
No	ITEM NAME	ITEM id	ATTRIBUTE	CONTENTS/MEANING/OTHERS (TRANSACTION CONDITIONS PART)		REMARKS

FIG. 39

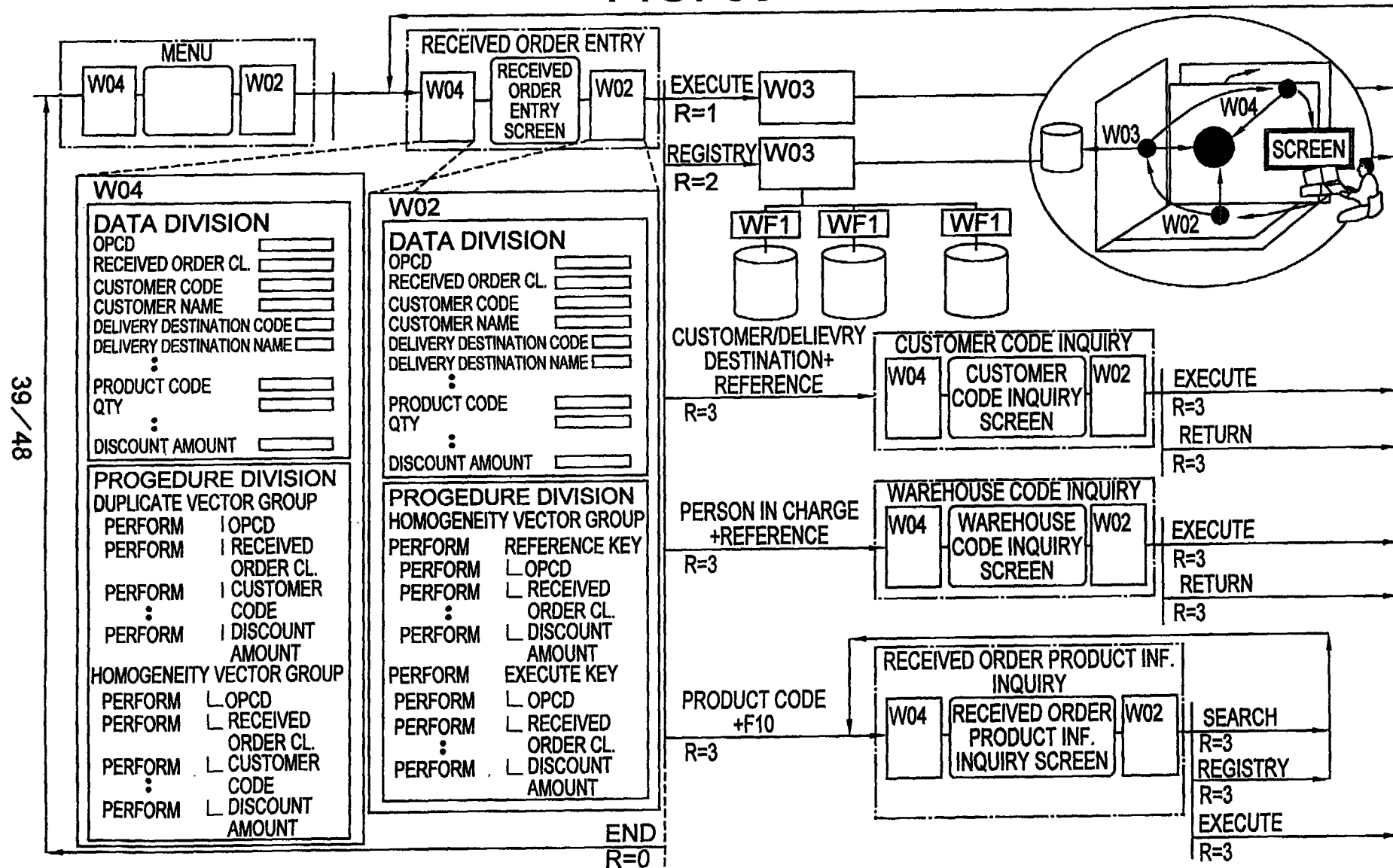


FIG. 40

DOCUMENTS FORMING ACTIVITY BASE

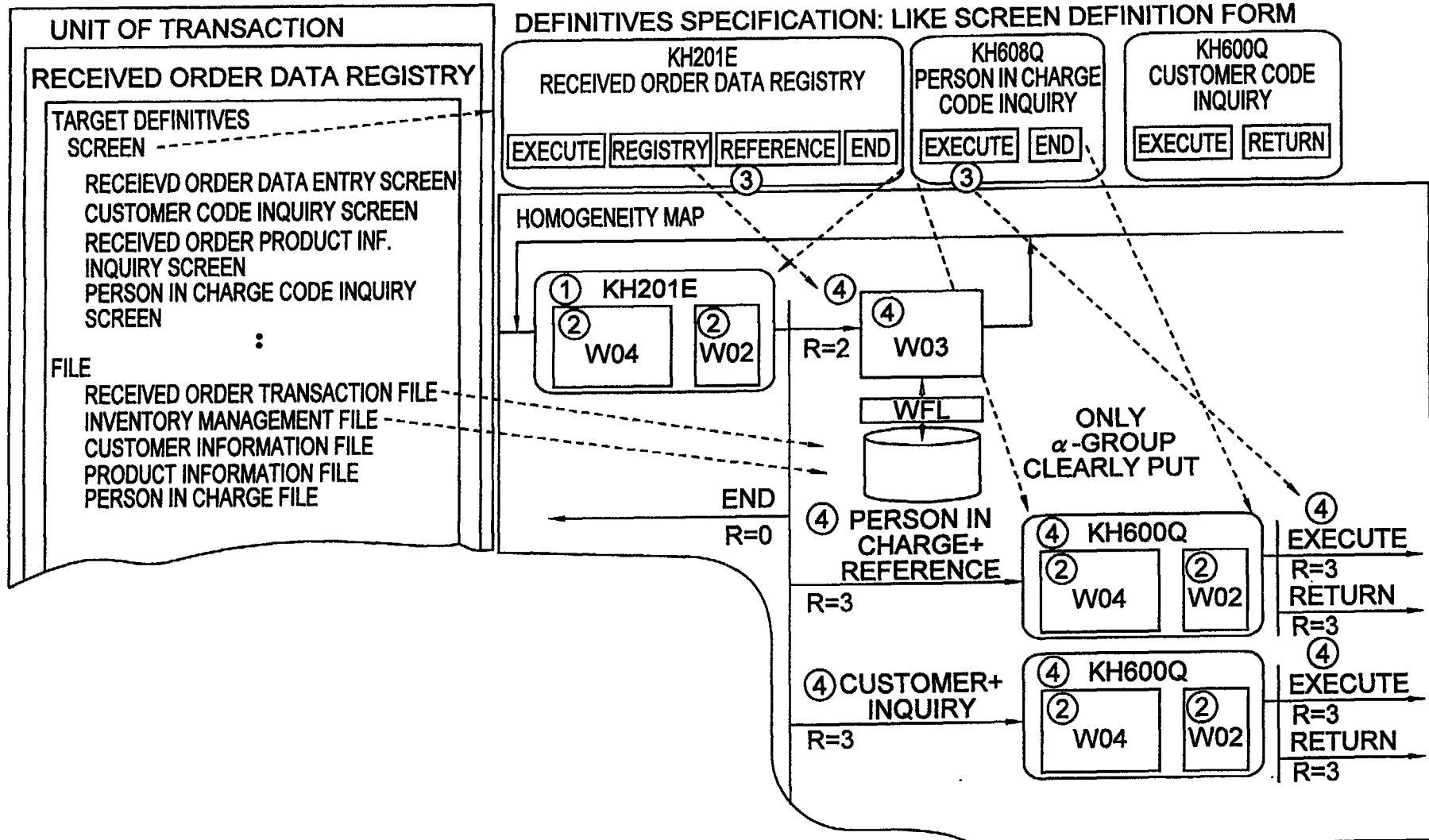


FIG. 41

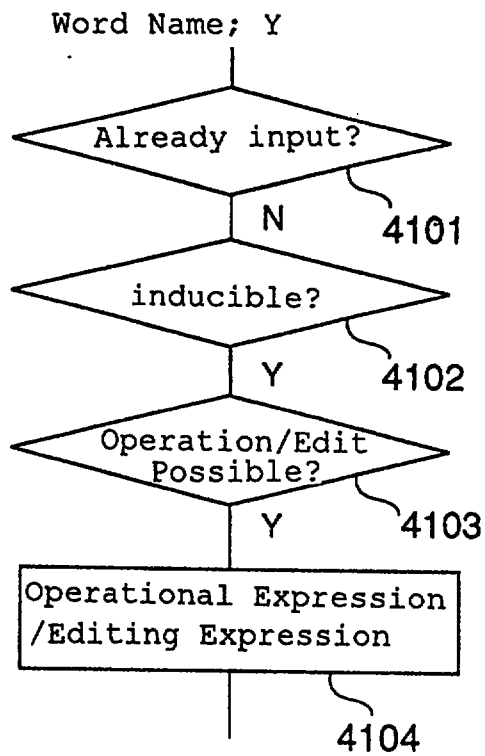


FIG. 42

DEFINITIVE NAME	MULTIPLE TRANSACTION FILE	DEFINITIVE id	KD52BP	PALLET CHAIN FUNCTION CL.	φ 0	AREA CL.	W03	PALLET id	W03KH210E	TENSE CONTROL VECTOR CL.	LOGIC
No.	ITEM NAME	ITEM id	OPERATIONAL SIGNIFICANCE CONDITIONS	OPERATIONAL EXPRESSION					REMARKS		
76	DISCOUNT TOTAL AMOUNT	KD1099	TRANSACTION CL. IS ORDER RECEIVE AND TRANSACTION CL. IS EXCEPT 1 (NO DISCOUNT) AND PRODUCT QTY OF EACH TRANSACTION IS NOT 0.	DISCOUNT TOTAL AMOUNT (KD1099) SPECIFICATION'S LINE NUMBER = Σ {TRANSACTION QTY UNIT PRICE DISCOUNT AMOUNT (KEC251)}					WHEN OPERATION IS ACCEPTED, THE OPERATION ACCEPTANCE FLG IS SET. WHEN THE TRANSACTION QTY UNIT PRICE DISCOUNT AMOUNT IS NOT CALCULATED, THE PALLET RESTART PROCESS IS EXECUTED.		
77	QTY UNIT PRICE TOTAL AMOUNT	KD1857	BUSINESS CL. IS ORDER RECEIVE AND PRODUCT QTY OF EACH TRANSACTION IS NOT 0.	QTY UNIT PRICE TOTAL AMOUNT (KD1857) SPECIFICATION'S LINE NUMBER = Σ {TRANSACTION QTY UNIT PRICE DISCOUNT AMOUNT (KEC201)}					WHEN OPERATION IS ACCEPTED, THE OPERATION ACCEPTANCE FLG IS SET.		
78	TRANSACTION TOTAL AMOUNT	KD1854	TRANSACTION CL. IS ORDER RECEIVE ONLY.	TRANSACTION TOTAL AMOUNT (KD1854) = QTY UNIT PRICE TOTAL AMOUNT (KD1851) - DISCOUNT TOTAL AMOUNT (KD1099)					WHEN OPERATION IS ACCEPTED, THE OPERATION ACCEPTANCE FLG IS SET. WHEN THE QTY UNIT PRICE TOTAL AMOUNT AND THE DISCOUNT TOTAL AMOUNT IS NOT CALCULATED, THE PALLET RESTART PROCESS IS EXECUTED.		
79	QTY PRICE TOTAL AMOUNT	KD1856	TRANSACTION CL. IS ORDER RECEIVE AND THE TRANSACTION PRODUCT IS NOT ORDER UNACCEPTABLE, THE TRANSACTION PRODUCT QTY IS NOT 0.	QTY PRICE TOTAL AMOUNT (KD1856) = PRICE (KE1124) + DISCOUNT PRODUCT QTY (KE5001)					WHEN OPERATION IS ACCEPTED, THE OPERATION ACCEPTANCE FLG IS SET. WHEN THE TRANSACTION PRODUCT IS NOT TRADABLE, MESSAGE cd=4101 IS RESPONDED, AND THE CORRESPONDING PRODUCT CODE FIELD IS DISPLAYED REVERSELY.		

FIG. 43

DEFINITIVE NAME	RECEIVED ORDER INPUT (DISTRIBUTION)	DEFINITIVE id	K201E	PALLET CHAIN FUNCTION CL	φ 0	AREA CL.	W04	PALLET id	LA1A1C40	TENSE CONTROL VECTOR CL.	LOGIC
No.	ITEM NAME	ITEM id	OPERATIONAL SIGNIFICANCE CONDITION	OPERATIONAL EXPRESSION					REMARKS		
56	TOTAL AMOUNT	A04360	PRODUCT QTY OF EACH TRANSACTION AND THE UNIT PRICE IS NOT <u>0</u> .	DISCOUNT TOTAL AMOUNT (AD4360) SPECIFICATION'S LINE NUMBER $= \sum \{ \text{TRANSACTION QTY UNIT PRICE AMOUNT (A16721)} \}$					WHEN OPERATION IS ACCEPTED, THE OPERATION ACCEPTANCE FLG IS SET. WHEN THE TRANSACTION QTY UNIT PRICE AMOUNT IS NOT CALCULATED, THE PALLET RESTART PROCESS IS EXECUTED.		
57	OPCD	A01540	UNCONDITIONAL	THE CORRESPONDING OPCD MUST EXIST IN THE PERSON IN CHARGE MASTER.					WHEN NOT EXISTING IN THE PERSON IN CHARGE MASTER, THE MESSAGE cd= 2001 IS RESPONDED, AND THE CORRESPONDING OPCD FIELD IS DISPLAYED REVERSELY.		
58	OPERATION PERSON IN CHARGE NAME	A01560	IN THE CASE THIS OPCD EXISTS IN THE PERSON IN CHARGE MASTER	OPERATION PERSON IN CHARGE NAME (A01560 OF KE201E) = OPERATION IN CHARGE NAME (SN0792 OF PERSON IN CHARGE MASTER)							

FIG. 44

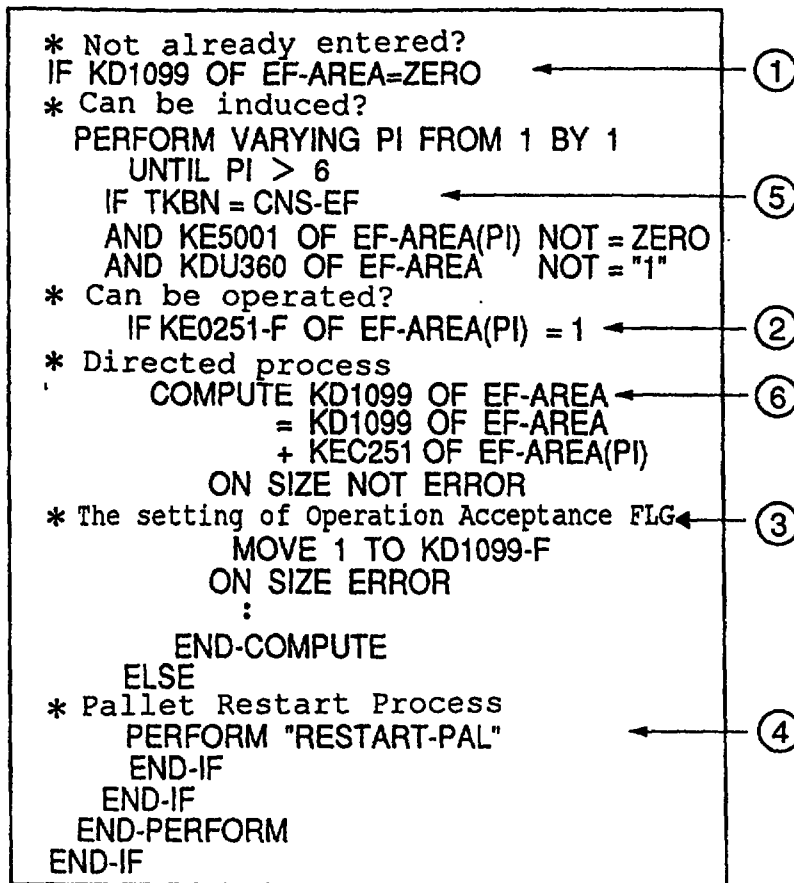


FIG. 45

Target Points	Traditional	LYEE	Effects
① Improvements in working method			Short-term development realized: 1/2 - 1/10
② Establishment of Quality Concept	Work logic by user Indispensable control logic	Work logic by user	Compression of the gross development volume: 20% - 80%
③ Exclusion of Empiricism	Dependent on empiricism	LYEE theory	Enhancement of the maintainability: 40 - 100 times.
④ Specialization of the Thinking Method	TOOL effects	TOOL effects Compression of dependency on personal abilities	Enhancement of the work efficiency: 40 - 100 times.

☒ 46

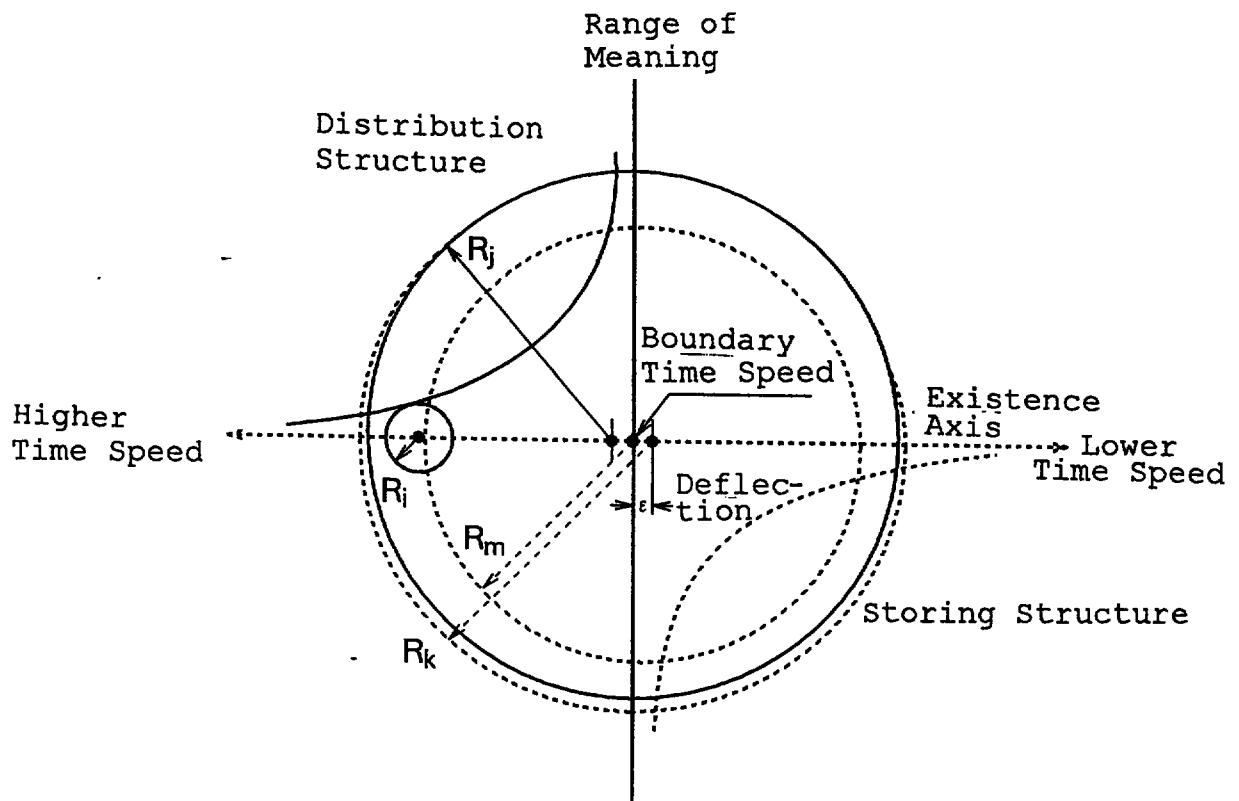
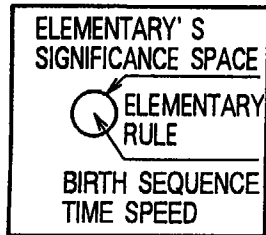
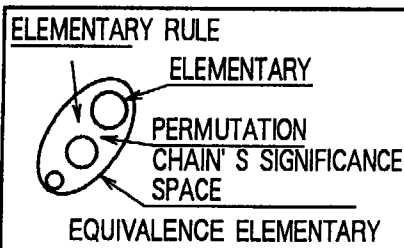


FIG. 47

STRUCTURE OF ELEMENTARY



STRUCTURE OF CHAIN



CONSCIOUSNESS CHAIN

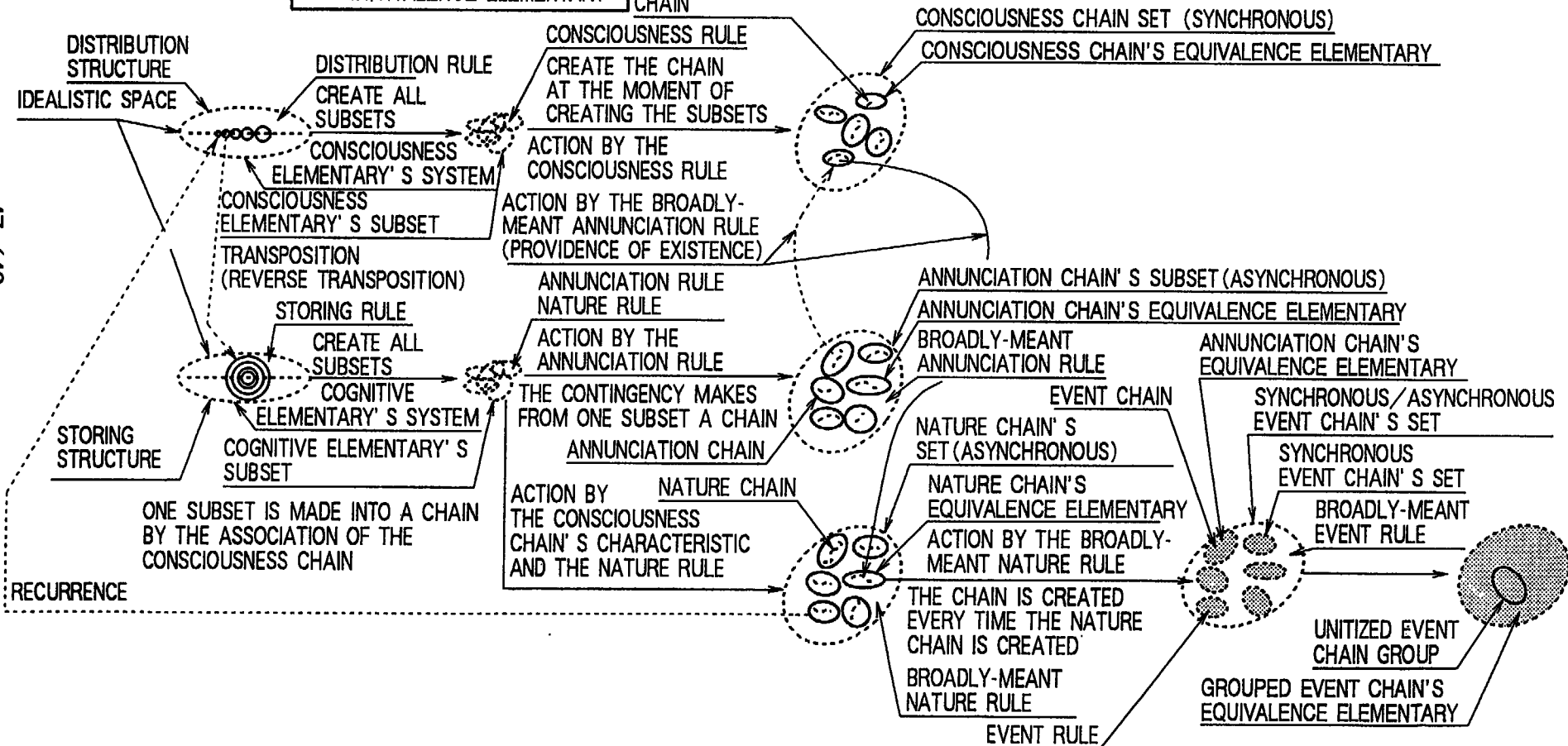


FIG. 48

