

Its structure is shown in Fig. 18.

11.  $\Phi 1$  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.

5

**[0043]** 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.

**[0044]** To proceed with practical works, a homogeneity map is produced.

10 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,  $T0 = \phi 0(\{f, 2\}k + \{g, 3\} + \{q, 4\}k)$

**[0045]**  $\{f, 2\}k$ ,  $\{g, 3\}$ ,  $\{q, 4\}k$  are tense control vectors, expressing a pallet not inclusive of T1.

15 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.

20 **[0046]** 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.

25 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.

**[0047]** The tense control vector is a general nomenclature of the duplicate vector and the homogeneity vector.

30 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.

**[0048]** 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.

35 **[0049]** 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.

**[0050]** 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.

40 **[0051]** 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.

**[0052]** 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,

45 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.

50 **[0053]** 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.

**[0054]** 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.

**[0055]** 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.

55 **[0056]** 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 mean-