

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).

**[0176]** 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).

**[0177]** 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.

**[0178]** 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

**[0179]** 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.

**[0180]** 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.

**[0181]** Explained next is the application range of Lyee in the system development step.

**[0182]** 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.

**[0183]** 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.

**[0184]** In the following, the working steps for the development of the Lyee-applied software is more specifically explained.

**[0185]** 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.

**[0186]** 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.

**[0187]** 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.