

block diagrams. As described above, if common information are mutually converted and reflected between different representation formats, inconsistencies between descriptions can be avoided.

Next, a data structure is described as shown in FIG. 54. A corresponding list (not shown) and an article list are described in a relation table 95, while an article name code and an article quantity are described in a data block diagram 96. In the example shown in FIG. 54, the corresponding list describes only the article name code and the article quantity which are inevitably necessary. The described data structure, the data arc on the functional block diagram and the signal lines on the sequence chart are mapped by a mapping manipulation. These mapping manipulation makes the user free from an annoying and complicated work of naming the articles.

Then, a definition of the detail of an internal module is executed as shown in FIG. 55. Referring to FIG. 55, an article name retrieval module 91a included in an object module to be described is a file manipulation and is hence described in a table handling diagram 96. In this table handling diagram 96, input/output information is converted and reflected by a functional block diagram that has already been described before. The structure of table data in table handling diagram 96 is cited from relation tables 95a and 95b. The cited data structure is mapped with an original relation table. Accordingly, if a data structure of the original relation table is altered, then a data structure in this table handling diagram 96 is also altered. This makes it possible to prevent a forgetting of alteration with the alteration of specifications.

In the article name retrieval module, its specification description is completed and then becomes executable by the description in the table handling diagram shown in FIG. 55. A prototyping is carried out for this executable article name retrieval module as shown in FIG. 56, to verify the specification description contents. More specifically, an article code of an article list 96a is set as input data, while a corresponding list 96b is set as output data. An article name code included in the article list is filled in the article code of article list 96a. A subsequent executing operation of the unit causes an article corresponding to this article name code 3 to be displayed on a corresponding list along with a corresponding quantity thereof. The specification description contents is altered on the basis of the result of the execution obtained by this prototyping and in accordance with an object specification, then executed again and verified.

If the description of the specification of the retrieval system is thereafter completed, the description is a single module that is partly completed and is registered as "parts".

Although the present invention has been described and illustrated in detail, it is clearly understood that the same is by way of illustration and example only and is not to be taken by way of limitation, the spirit and scope of the present invention being limited only by the terms of the appended claims.

What is claimed is:

1. A program development apparatus, comprising:

specification description means for producing a graphic specification description with respect to an object program on a display screen using a non-text style and graphically visual language, said specification descrip-

tion means including describing means for describing a specification using a plurality of non-text style and graphically visual languages different from each other to produce said graphic specification description;

conversion means for converting the specified description produced by said specification description means written in at least one of said non-text style and graphically visual languages into an executable program; and

file management means for storing therein as files, a plurality of data indicating the specification description produced by said specification description means in at least one of the non-text style and graphically visual language formats and a plurality of data indicating the program converted by said conversion means.

said file management means including:

a memory medium for storing therein said plurality of data together with a plurality of identifiers associated with a data structure, and

manipulation apparatus configured for managing a file in accordance with a primitive file manipulation instruction so as to hide details of an operation on a data structure stored in the file, responsive to an applied instruction for manipulating the structure of the data stored in said storage means by employing said identifiers.

2. The program development apparatus according to claim 1, further comprising,

parts management means including means for registering as parts information, the specification description produced by said specification description means, and means for exhibiting the registered parts information.

said file management means further storing therein as files, a plurality of data indicating the parts information registered by said registration means.

3. An operating method of a program development apparatus for managing a file in accordance with a primitive file manipulation instruction so as to hide details of an operation on a data structure stored in the file, comprising the steps of:

producing a graphic specification description with respect to an object program on a display screen using different non-text style and graphically visual languages;

converting said produced graphic specification description into an executable program;

storing a plurality of data indicating said specification description in the used non-text style and graphically visual language formats and a plurality of data indicating said executable program, together with a plurality of identifiers; and

manipulating a data structure in response to an instruction specifying said identifiers.

4. The operating method of the program development apparatus according to claim 3, further comprising the steps of:

registering said produced specification description as parts information;

storing a plurality of data indicating said registered parts information together with an identifier; and

manipulating the structure of said stored data by employing said identifier in response to an applied instruction.

* * * * *