Automatic connection of specification documents and software via ADLib_W

Koichi Ogaki
Yonghwa lee
Eiji Hojo
Akira Ikezoe

In recent times design of power train control ECUs has basically consisted of combining software part modules (abbreviated to "modules" below), and standardization / modularization of specification documents and software has been undertaken for use in constructing environments in which the software can be finished into a design product. For diagnosis, one of the major forms of control implemented by power train control ECUs, the objects of control are clearly laid down in legal regulations and the modules necessary can be determined once the sensors and actuators to be installed in the vehicle are given.

Selection and connection of modules used to be performed by manual work and entailed expending large amounts of time in order to maintain design quality. Accordingly we are developing a database in which the design know-how acquired by our company over many years of power train control ECU design can be accumulated as intellectual property (abbreviated to "IP" below), and furthermore are starting development of a system called an automatic connection tool.

This paper provides an introduction to the ADLib_W tool for automatic connection of specification documents and software in diagnosis that is under development.

ADLib_W: Assistant-Design System by Data Base Library for W ("W" being a development code for diagnosis in ADLib)

1 Introduction

In order to create an environment where product software is completed basically by combining modules, recent power train control ECU designs have promoted the standardization and modularization of specifications and software.

In other words, the key is to select and combine optimum modules for the required specifications without error, from a wide variety of standard modules. (Refer to Fig. 1.)

Diagnosis is one of the major forms of control implemented by the power train control ECU. When failures occur in the sensors installed on the vehicle, or an internal error occurs in the power train control ECU, it lights warning indicators to alert the driver to avoid danger, or displays vehicle status on specified tools to improve serviceability. The subjects of diagnosis control are stipulated by legal regulations, and design errors are not allowed. For this reason a large amount of time is spent for specifications and design at our company, and is a subject which we strongly desire to automate in order to maintain quality and increase design efficiency.

A necessary condition for automation is that specific rules must be established for the selection of modules.

This makes diagnosis very suited for automation, since module selection for it are made based on legal stipulations.

Optimal modules are selected from a standard module library according to the required specifications

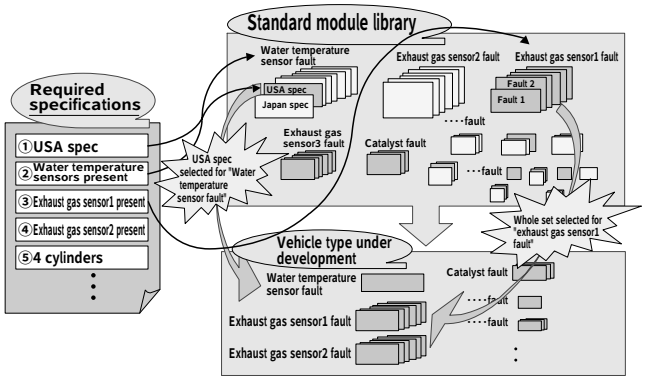


Fig.1 Software design in recent times

2 Purpose of the development

Our company has been engaged in the design of power train control ECUs for many years, and has steadily accumulated knowledge in the field. Now we are converting our knowledge for diagnosis related software design into databases, in order to accumulate it as intellectual property (IP), and have also started development of a system called an "automatic connection tool."

In order raise efficiency and quality, we have always been actively engaged not only in the development and production of ECUs, but also in the development of tools, utilizing our accumulated technical abilities. And we have been providing design support tools to support customers in their work.

The main purpose of this new tool is to cut the manhours required for design through the automation it provides, but in addition it has been given ample consideration for enhancement of design quality.

The flow of design work (excluding evaluation processes) in our software design department has traditionally taken the form described below.

①Development of standard modules - Implemented on an irregular basis

The software modules required in the design of software for various vehicle types are designed in accordance with the specification documents obtained from the customer, in such a way as to yield flexibility. After their design has been completed, the modules are registered in a library of standard modules.

②Deliberation of required specifications - Implemented during vehicle type design

From information gained from sensors and actuators to be installed on the vehicle, required specifications which match the legal regulations are considered, then modules that should be connected and other setting values that are required for the connections are determined.

③Connection of software - Implemented during vehicle type design

The modules specified by the specifications are connected according to a design procedure that is laid down in great detail in order to secure quality.

The design working procedure is laid down in detail in order to prevent problems. This secures quality but entails expending large amounts of working time.

By utilizing the new ADLib_W tool to be developed, it will be possible to maintain the high quality which was ensured by large amounts of work hours while greatly optimizing efficiency of processes ② and ③ above.

②Deliberation of required specifications with ADLib_W - Implemented during vehicle type design

By feeding information gained from sensors and actuators on the vehicle into the ADLib_W, the module to be connected and the other setting values required for the connections are automatically selected and fixed.

③Connection of software with ADLib_W - Implemented during vehicle type design

By feeding information gained from sensors and actuators on the vehicle into the ADLib_W, the module is automatically selected, and connected.

3 Product overview

Diagnosis is one of the major forms of control implemented by the power train control ECU. When failures occur in the sensors installed on the vehicle, or an internal error occurs in the power train control ECU, warning indicators are lighted to alert the driver to avoid danger, or vehicle status is displayed on specified tools to improve serviceability. The subjects of diagnosis control are stipulated by legal regulations of each country.

Because of this:

①Depending on the settings of the I/O (input/output signals of the ECU, in other words, whether sensors / actuators are installed on the vehicle), destination (i.e. the legal regulations of country of sale) and other supplementary data (termed "system data" below), the specifications that must be included are determined automatically.

②The specification documents and software are being standardized and divided into modules.

Automation will be achieved by utilizing these two characteristics, relating "I/O, system and destination" with "standardized modules", and by creating a database. (Refer to Fig. 2.)

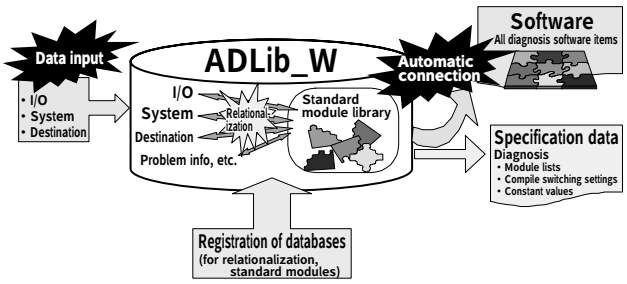


Fig.2 Overview of ADLib_W

∗Languages used: VB, VC++
∗Databases used: ACCESS and ORACLE
∗Operating environment: Windows

4 Development technology

4.1 Process Flow up to the Automatic Connection

When the I/O, system and destination data for the vehicle to be designed are entered to the ADLib_W, the tool automatically selects standard modules matching the input data, according to the relational database. Furthermore, the function call data and constant data required for automatic connection for each standard module, are registered in the ADLib_W as a database. The various specifications and software items are automatically connected based on these databases. (Refer to Fig. 3.)

(1) Initial registration

Initial registration is performed for a vehicle that is to be developed. When multiple vehicles are to be designed simultaneously, the number of vehicle variations and the destinations matching each vehicle variation are registered. (Refer to Fig. 4.)

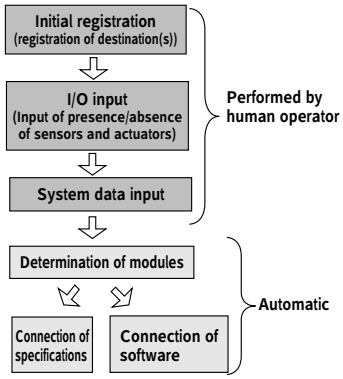


Fig.3 Flow of automatic connection

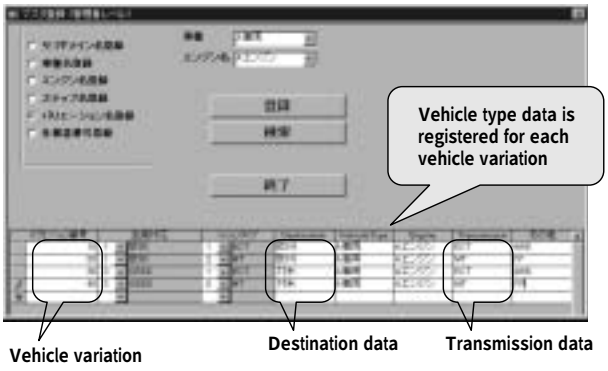


Fig.4 Initial registration screen

(2) I/O and system data input

I/O and system data matching each vehicle variation are selected. (Refer to Fig. 5.)

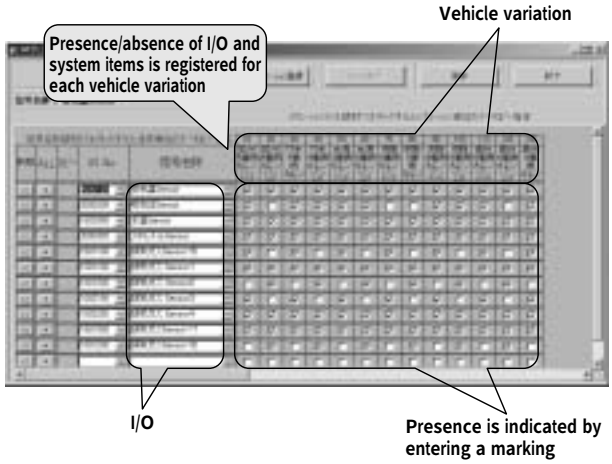


Fig.5 I/O / system data inputting screen

(3) Determination of modules - Automatic

Next, modules matching the vehicle are drawn from the database. This is performed automatically according to the results of the I/O and system data selection in step (2), and the modules determined for each vehicle variation receive an application marking. (Refer to Fig. 6.)

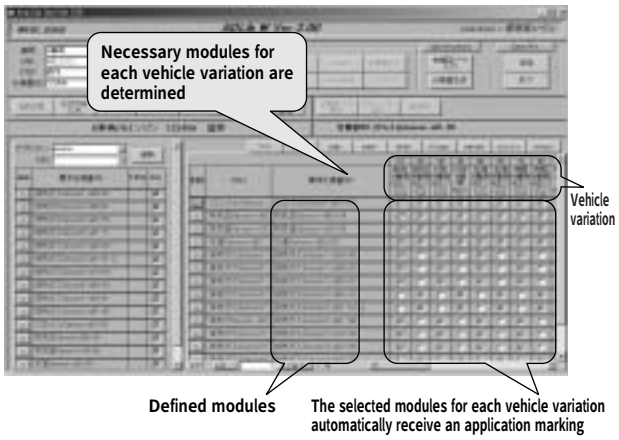


Fig.6 Module determination screen

(4) Connection of specifications - Automatic

The specifications are automatically connected based on the I/O and system data input information and module determination information.

Specifications are broadly divided into 3 categories:

①"Module list" which specifies the modules in the standard module library that are to be used for the present design;

②"Constant values" which indicate whether or not the features which monitor "matching values gained from vehicle test results" and "operational conditions of individual sensors gained through an external tool" exist on the software;

③"Compile switching setting values" which indicate the parameter settings matching the vehicle that the software with built in parameter switching is to be developed for.

①Module lists

The modules to be used for the present design are selected automatically via the module determination in step (3). The module lists take the form of the standard module library listed in specification document format with application markings assigned to the modules to be used.

②Constant values

The matching values are excluded as subjects of automatic connection since they are determined via tests and cannot be derived solely by a series of relations. On the other hand, the settings indicating whether or not monitoring functions are to be employed, are determined by the I/O and system input data. Constant values matching the input data are derived via the relational database that is registered within the ADLib_W and are compiled in a specification document format.

③Compile switching setting values (for internal condition switching in the software)

Compile switching is determined by relating the I/O and system input data.

For such switching therefore, setting values conforming to the input data are derived via the relational database that is registered within the ADLib_W and are compiled in a specification document format.

(5) Connection of software - Automatic

Based on the I/O, system data input and the module determination information, software is connected automatically. (Refer to Fig.7)

Software is broadly divided into 4 categories: ① "Individual Modules" consisting of modules selected from the standard module library for use in the present design, ②"Function calls" which enable (call up) functions present within the modules selected, ③"Constant values" which sets on the software the applicable values derived from test results on the vehicle, and whether or not the features which monitors operational conditions of each sensor using an external tool exists, and ④ "Compile switching setting values" which specify condition settings for software incorporating condition switching in the interest of versatility (such settings must match the vehicle under development).

①Individual Modules

The modules to be used for the present design are selected automatically through the module determination in step (3). The selected module is automatically compiled from the standard module library. Individual modules compiled here is used as a base to perform coding for ② to ④ below.

②Function calls

Function calls must be added in order to enable the functions present within the modules. The coding for this is implemented automatically through databases for call sequences and call times, etc., that are registered in ADLib_W.

③Constant values

The matching values are excluded as subjects of automatic connection since they are determined via tests and cannot be derived solely by a series of relations. The settings indicating whether or not monitoring functions are to be employed, on the other hand, are determined by the I/O and system input data. For such settings therefore, constant values matching the

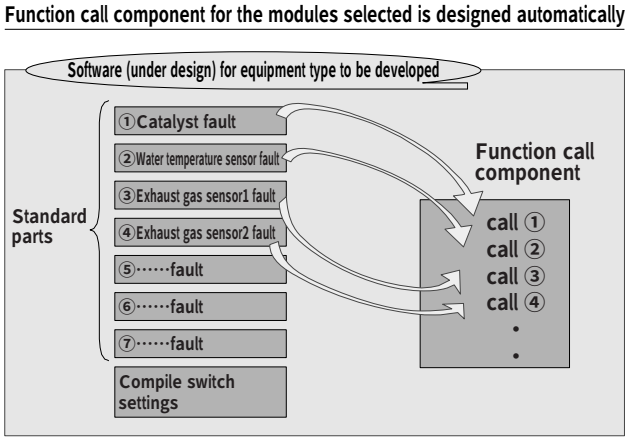


Fig.7 Conceptualization of automatic connection

input data are coded automatically via the relational database registered within ADLib_W.

④Compile switching setting values

Compile switching is determined by relating the I/O and system input data.

For such switching, setting values matching the input data are coded automatically through the relational database registered within ADLib_W.

4.2 Construction of database for realizing automatic connection

We constructed the relational database to relate the ①"I/O, system and destination" with ②"standard modules" such that ① is selected automatically when ② is entered. (Refer to Fig. 8.)

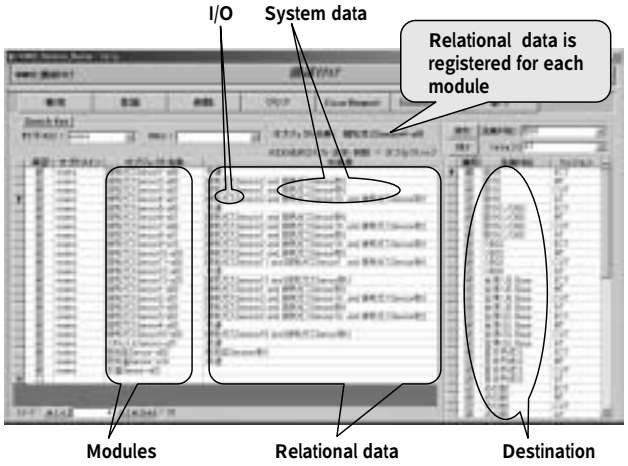


Fig.8 Module relationalization database

4.3 Resolution of problems in automation

1) Improving database maintainability

Because the standard modules required for construction of the database in 4.2 are frequently added and changed, the maintainability of the database is of considerable importance as regards its utilization for daily operations. The problem of improving the database maintainability in terms of registration, etc., of standard modules was resolved by the following methods.

Additions and changes to the standard modules were being performed by manual input to the ADLib_W database in accordance with a "Standard module list sheet" issued each time such updating was necessary. To simplify this step we had the list sheets prepared as Excel data in a fixed format, and incorporated a function for automatic input of such data into the ADLib_W database. Thanks to this the user has only to click on an "Incorporate list sheet" button to have the standard module database automatically altered to the latest settings.

2) Improving convenience of system data entry (The item under examination)

The number of system data items that had to be input when designing a vehicle type amounted to be over 600 items, so in order to reduce the man-hours required by automation it is necessary also to improve the convenience of data entry. Accordingly we are considering the following method for reducing the number of data items, which will make it possible to reduce the number of items that must be entered manually to around 1/3 (i.e. 200 items) compared to the previous system.

①The input items are arranged into a 2-layer hierarchy consisting of categories and the parameters within those categories.

②System Items that were not deemed applicable in the I/O input and items linked to them are automatically set as being not applicable. Therefore the all items existing below a large group that is not required will no longer require entry, thereby making it possible to greatly reduce the number of items required for entry.

③Items deemed applicable in the I/O input and system items linked to them will require entry to lower level items. However, in order to reduce the number of entries, system items that are linked with the I/O are set automatically as being applicable.

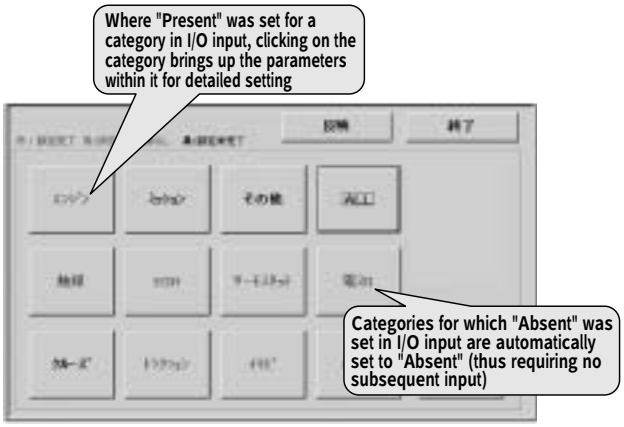


Fig.9 System data inputting screen: categories

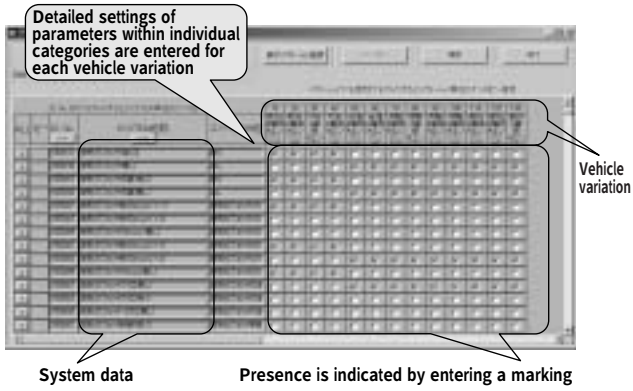


Fig.10 System data inputting screen: parameters within categories

3) Flexibility

Totally automating the tool would have rendered it unable to handle exceptional specifications and thus have resulted in an unusable tool. Accordingly the tool

has been made basically automatic but is also configured to accommodate manual changes where necessary. It is further configured such that manual changes to relationalized data content will be automatically reflected in subsequent automatic connection.

6 Conclusion

This concludes the description of the ADLib_W tool, for automatic connection of software and specifications in diagnosis.

Our company has continually worked in improving efficiency and design quality by promoting the introduction of tools in various processes of software design. Development of tools has in the past centered on the downstream processes (evaluation processes) but the introduction of ADLib_W will permit extension of the tool chain to upstream processes (specification deliberation and design processes). We believe this represents a major step forward for the use of tools to create a seamless software development environment, extending from the consideration of specifications through to the evaluation.

The following endeavors are planned for the future.

①Various forms of design know-how stored on paper medium will be integrated as IP into ADLib_W.

②Tool chains will be joined up via linkage with the company-wide process management system, so as to cover processes from order receipt through to shipment. (Refer to Fig.11.)

③ADLib_W methodology will be applied to turn know-how in other fields into IP and provide automatic connection for such.

Currently ADLib_W is in a prototype stage, on its the way to being brought into practical use for mass production operations. Detailed improvements will have to be made to enable its application, but in the near future it is anticipated to bring reductions of 50% or more in the man-hours required for design work while maintaining design quality at our traditionally high levels. We also envisage that use of ADLib_W will bring about a transformation to a new style in our work (Refer to Fig.12.). Thus, great things are expected of this tool in the future.

Finally we would like to express our sincere gratitude to all those within and without the company who have cooperated in the development of ADLib_W.

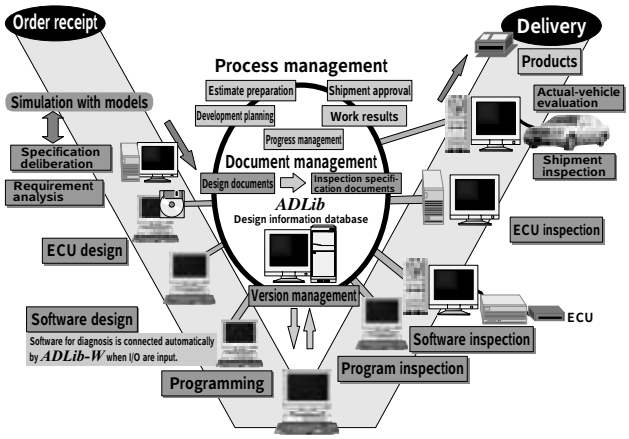
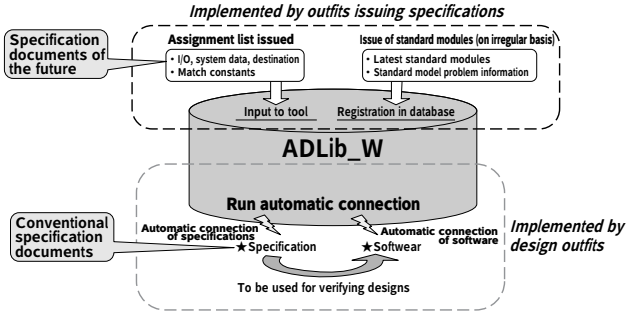


Fig.11 Software development process and construction of environments



With ADLib_W, the data for only the I/O, system and destination data are reequired; the specification documents that have traditionally been employed will be used to verify designs. Furthermore, ADLib_W does not merely constitute automation - through sharing of the design input data on electronic media it can also be expected to provide even greater design efficiency and design quality.

Fig.12 Projected conceptualization of the "style" of design using ADLib_W

<ON Trademarks>
VB…Microsoft® Visual Basic®
VC++…Microsoft® Visual C++®
ACCESS…Microsoft® ACCESS
ORACLE…ORACLE®
Windows®…Related Microsoft® Windows® 95, 98, NT, 2000

Profiles of Writers

Koichi Ogaki

Entered the company in 1991. Since then, has pursued development of software for electronic automobile engine control equipment. Currently in the Software Engineering Department of the Vehicle Electronics Products Group.

Yonghwa lee

Entered the company in 2001. Since then, has pursued development of software for electronic automobile engine control equipment. Currently in the Software Engineering Department of the Vehicle Electronics Products Group.

Eiji Hojo

Entered the company in 2000. Since then, has pursued development of software for electronic automobile engine control equipment. Currently in the Software Engineering Department of the Vehicle Electronics Products Group.

Akira Ikezoe

Entered the company in 1982. Since then, has pursued mechanism design for electronic automobile engine control equipment, software development, and other projects. Currently the Department General Manager of the Software Engineering Department of the Vehicle Electronics Product Group.