Software System Integration Support Expert System
Based on the Object-Oriented Method

Michio Ito, Shinpei Okemura and Osamu Okuzawa
Network Information Systems Laboratories
NTT
Yokosuka-shi, Kanagawa, Japan

Shoji Takahashi
Network System Development Center
NTT
Chiyoda-ku, Tokyo, Japan

Abstract
To make the best use of computer systems, it is necessary to establish optimum initialization parameter values according to computer system configuration and the purpose of use. However, a very high-level of technical knowledge is required for software system generating (SG) work. Therefore, it is very important to simplify SG work in order to popularize computer use. Towards this purpose, we have developed an SG support expert system designated SG-BOY for NTT general purpose computer systems. This system features an object-oriented knowledge base to make maintenance easier. This paper discusses the SG-BOY system structure and an application evaluation of the system.

1 Introduction
For large computer systems, general-purpose control programs such as operating systems, communication control programs and DB/DC management programs provide effective and easy ways to use the systems. By merely combining suitable control programs and adding application programs, user can freely use computer systems for various purpose. However, to make the best use of many general-purpose control programs, system generating (SG) work is indispensable. SG work involves specifying correct parameter values and integrating them into software systems. To execute SG work without error, advanced technical knowledge is required for each control program. SG work is a bottleneck to the mass installation of computer systems because of the shortage of specialists. SG work also involves several other problems due to the need to perform the work manually. For example, many working hours are required, and human error occurs frequently.

To solve such problems, we propose to develop an expert system having the following functions:
- Contains the technical knowledge related to the initialization information for each control program in a knowledge base, and
- Supports accurate SG work with quick, simplified input.

However, no previous development of similar expert systems has been previously reported. The closest is the XCON system developed by DEC, which assists in configuring hardware systems [1]. Half of the XCON knowledge base is changed every year, and improvement of maintenance is the most important item of concern remaining [2]. To simplify SG work for NTT general-purpose computer systems (DIPS), we developed an expert system (SG-BOY) that optimizes SG work for control programs. This system is probably the first SG support expert system in the world. It can be used directly by end users as well as SG specialists. It adopts object-oriented knowledge structures to make maintenance easier. By using this system, the time required to perform a given amount of SG work can generally be reduced from one month to only one day. This paper reports on the system structure, processing method, and application evaluation of the SG-BOY system.

2 System Outline
2.1 SG-BOY Functions and Design Policies
System generating (SG) work for NTT general-purpose computers (Denden Information Processing System, or DIPS) is divided into two phases as shown in Figure 1.

- Phase 1 (SG control statement design): in this phase, the SG control statements that specify the program configuration conditions are designed according to the desired system conditions.
- Phase 2 (SG execution): in this phase, SG tools install the control programs into the computer by using the SG control statements.

Phase 1 involves 90 percent or more of the work. For a complicated system, it usually takes two to four weeks for a
specialist to execute this phase work.

To increase the efficiency of this phase work, the SG-BOY expert system makes it possible to generate SG control statements quickly and accurately according to the desired configuration conditions merely by inputting very simple data.

We developed SG-BOY based on the following main design policies:

For users:
(1) To simplify the input system so that users who have no special knowledge can use it (e.g., by decreasing the number of input items and by guidance with selection menu)

(2) To increase the reliability of SG work by means of preventing design errors caused by manual operation (e.g., by interactive input data validation and automatic generation of correlative parameter values)

(3) To reduce the SG work time (e.g., reduction of work time for writing and reviewing documents)
For system structure:
(1) To adopt object-oriented design and structure enabling easy maintenance, since the objective control programs may be extended and changed frequently.
(2) To enable the expert system to be operated on a personal computer that can be easily prepared and directly operated by users.

2.2 System structure

SG-BOY operates on a personal computer. It is written in LISP using KBMS/PC (Knowledge Base Management System/PC) [3], which is an expert system construction support tool developed by NTT. As shown in Figure 2, SG-BOY is composed of the following three subsystems:

- Knowledge base management and execution subsystem
- Human-machine interface (HUI) control subsystem
- Output control subsystem

3 Knowledge Base Structure and Processing Method

3.1 Improving knowledge design and maintainability with 3-level hierarchical structure

SG-BOY generates SG control statements based on input from the operator. Therefore, the following types of knowledge are necessary for this expert system:

- Knowledge on how to store the input conditions correctly (I: input knowledge)
- Knowledge on how to generate SG execution control statements (II: SG control statement knowledge)
- Knowledge on how to associate the input knowledge with the SG control statement knowledge (III: SG design knowledge)

For SG-BOY, we aimed at improving and enhancing design and maintenance by adopting the object-oriented design method. With this method, the types of knowledge above are designed separately and managed in a 3-level hierarchical structure.

a) Easy design: Knowledge types I and III can be handled as independent objects. Both types can be designed independently and in parallel. Knowledge type II can be designed automatically if I and III have been determined by SG design specialists.

b) Easy maintenance: For knowledge types I and III, the 3-level hierarchical structure lets additions and other changes be made independently only within the related objects. For knowledge type II, the correspondence between I and III is defined clearly.

3.2 Structure of condition input knowledge

There are two methods to determine managed objects related to the condition input knowledge, depending on the purpose:

- To simplify input by the operator, the managed objects are determined based on the terms and conditions of the specific system (e.g., names and numbers of connected workstations and switching equipment).
- To attach importance to general use, the managed objects are determined on the basis of the conditions of the computer system and control programs subject to SG (e.g., machine type of the computer and database file.

![Figure 2 SG-BOY System Structure](image-url)
For SG-BOY, we determined the managed objects and designed the knowledge structure from the viewpoint of simplifying input. To simplify operation, it is most important to determine the system configuration conditions with minimum input information. For this purpose, independent input objects are selected from the system configuration conditions. The input frame is then defined, treating the set of attributes of each object as one class. A condition which must be input repeatedly is stored in the knowledge base as an individual instance of the same class. In addition, input is further simplified by internally generating the items (serial numbers, symbolic identifiers, etc.) that can be automatically determined in the system.

SG design errors can be prevented if input information is thoroughly checked. For this purpose, the input check frame is defined for the condition input knowledge. The input is temporarily stored in this frame. The validity of input frame is checked by rules and functions for input error checking. The input frame is stored after validity has been assured. To enable high-speed processing, the input check frame is placed into the memory using the working memory function of KBMS.

3.3 Simplifying the knowledge structure for generating SG control statements.

The knowledge structure for generating SG tool control statements is very simple. This knowledge consists of the following:
- Control statement frames in which the control statement type of each SG tool (provided by the control programs) is defined as an object.
- Functions that output actual SG control statements for each instance of the frames.

The one problem related to this knowledge structure is that there are several thousand SG control statement types. If all the types are made into frames and functions, they exceed the processing limits of a personal computer. (The limit is assumed to be several hundred frames.) For this reason, we simplified the structure by making a fixed SG control statement patterns: Fixed patterns of SG control statements are prepared as prototypes. Only the generating knowledge related to changing control statements is made into frames and functions. These are combined with the prototypes. By adopting this method, the knowledge base for generating SG control statements was simplified to a great extent. The number of frames and functions used is about one-tenth the number of control statement types.

3.4 Structure of SG design knowledge

Usually, SG work needs to be entrusted to specialists. This is only because general users do not have the design knowledge to associate the input conditions with SG control statements. In other words, SG work can be executed correctly by general users, if this design knowledge has been incorporated in a knowledge base. For SG-BOY, a team of SG specialists compiled and arranged the design knowledge. The knowledge was reported to the expert system development specialists (knowledge engineers) who made this knowledge into a knowledge base. In general, the knowledge structures associating the input with SG control statements have a relationship of n to m. The knowledge involves conversion of attributes, numeric data, identifiers, and so forth. For actual construction, the following two patterns were used depending on the case:
- Rule type: A rule directly updates a control statement frame according to the input frame condition.
- Function type: A rule calls a function that updates an instance of a control statement frame.

3.5 Knowledge structure and processing flow

Figure 3-a shows the 3-level hierarchical knowledge structure of SG-BOY and its processing flow. The basic processing flow of SG-BOY is as follows:

a) Input from the operator is stored in the input check frame. The input check rules check the input for errors. Then, the input is stored in the input frame.

b) The rule of design knowledge is started. and instances are generated for each frame of control statement knowledge.

c) The control statement output function is started. Then, control statements are generated and combined with prototype control statements, and the final SG statement file is output to diskettes.

Figure 3-b shows the process with which the input node address information is reflected in the control statement. It also shows an example of an SG design knowledge rule that generates a control statement frame from the input frame.

4 Application Evaluation

SG-BOY is now being applied to several maintenance systems for NTT communication networks. The following describes the evaluation results based on field applications.

(1) Serviceability. It takes about two to four weeks for specialists to perform the SG design work for DIPS. It is necessary to design a large number of SG control statements involving thousands of steps. Accordingly, human errors, such as coding errors and careless mistakes, occur frequently. The reliability is also low. By using SG-BOY, SG work can be executed correctly in one or two days. It is only necessary for a non-specialist to design conditions for
about 100 items and to input the conditions on the screen for a personal computer. Network maintenance systems are installed to maintain and monitor NTT communication networks throughout the whole country. SG-BOY reduced the time required for changing these system configurations, and increased their reliability. Thus, it improved network serviceability.

(2) **Maintainability.** For upgrading a system, input conditions and specifications of SG control statements must often be added and modified. We adopted the object-oriented 3-level hierarchical knowledge structure. One of the most important reasons we used this method was to facilitate the maintenance of SG-BOY itself. Within one year after the development, we made modifications of about ten items for each pattern. With the object-oriented design, modification ranges can be localized, and maintenance can

![Diagram of 3-Level Hierarchical Knowledge and its Processing Flow](image)

**Figure 3-a** 3-Level Hierarchical Knowledge and its Processing Flow

![Knowledge Examples](image)

**Figure 3-b** Knowledge Examples
be carried out through easy modifications of rules and functions.

(3) Processing Performance. The knowledge base scale can be represented by the total number of rules, frames and functions. The SG-BOY knowledge base scale is about 500. Adopting the patternizing method limited the knowledge base scale to an appropriate range. Therefore, this system operates at a satisfactory processing speed on 32-bit personal computer.

5 Conclusion

We made the technical knowledge required for SG work into a knowledge base, and developed an expert system that can be easily operated using a personal computer. We proved that this system enabled an operator having no special knowledge to correctly execute SG work in a short time. Maintenance is facilitated by the object-oriented hierarchical knowledge structure design. Development of SG-BOY succeeded because of close cooperation between SG design specialists and knowledge engineers. In the future, we intend to expand the applicable range of SG-BOY by extending the patterns and knowledge.

Acknowledgements

We thank Dr. H.Ishino and Mr. M.Haishara at the Network Information Systems Laboratories, and Mr. K.Ohashi at the Information System Headquarters.

References