Journal of the Korean Nuclear Society
Volume 31, Number 1, pp.58~67, February 1999

The C Language Auto-generation of Reactor Trip Logic Caused by Steam Generator Water Level Using CASE Tools

Jang-Yeol Kim and Jang-Soo Lee
Korea Atomic Energy Research Institute
150 Dukjin-dong, Yusong-gu, Taejon 305-353, Korea

(Received May 26, 1997)

Abstract

The purpose is to produce a model of nuclear reactor trip logic caused by the steam generator water level of Wolsong 2/3/4 unit through an activity chart and a statechart and to produce C language automatically using Statechart-based Formalism and Statemate MAGNUM toolset suggested by David Harel Formalism. It was worth attempting auto-generation of C language though we manually made Software Requirement Specification(SRS) for safety-critical software using statechart-based formalism. Most of the phases of the software life-cycle except the software requirement specification of an analysis phase were generated automatically by Computer Aided Software Engineering (CASE) tools. It was verified that automatically produced C language has high productivity, portability, and quality through the simulation.

1. Introduction

For the successful digitalization of instrumentation and control systems, safety-critical software development, automatic programming and its V&V technology development through verified tools, for example, Computer Aided Software Engineering (CASE), are urgently required. Until now, the software has been developed from skilled software programmers. Hence, system analysis, development methodologies and coding methods produced by different programmers, have resulted in inconsistencies of software products. This situation has also caused difficulties in maintaining the software programs[5].

To resolve these problems, many efforts have been made by introducing information engineering methodologies. For example, the establishment of software development procedures, Standard Handbook Procedure(SHP) and CASE tools have all experienced excellent results in the reduction of software development time and the enhancement of quality software products[6].

In the case of the safety-critical software installed in reactor protection systems, software safety is very important and needs to have a verification process. In order to guarantee software safety and verification, proper development methodology and CASE tools are mandatory.

In this thesis, we selected the reactor trip logic

of the Wolsong 2/3/4 unit steam generator water level as an experimental model and performed the analysis and design modeling by Activity chart and Statechart and Graphic User Interface(GUI) panel and automatically produced C language. The development procedure was then verified.

Section 1 provides an introduction of C language auto-generation of safety-critical software. Section 2 includes automatic programming of safety-critical software. It is divide into six major subsections. Which are requirement analysis of nuclear reactor trip by steam generator water level, modeling procedure and C language auto-generation and execution. The modeling procedure comes from environmental modeling to module chart design. After the modeling procedure is completed, C language is automatically produced using CASE tools. Section 3 addresses conclusions of this paper.

2. Automatic Programming of Safety-critical Software

The software product engineering process consists of two models at each level in software development life cycle, which are the essential model level and the implementation model level. The essential model is the product of the analysis process. The implementation model is the product of design process[4].

The essential model consist of an environmental model and a behavioral model. An environmental modeling defines the environment of the Man-Machine Interface System (MMIS) design and the interface between the MMIS and the environment. The behavioral model is the last step for an essential model. The behavioral model describes detailed functionality in the MMIS.

As was noted above, the implementation model is the product of the design process. At the level of the implementation model, the system designer is primarily trying to decide how the essential model should be allocated to different functions which functions should communicate with one another.

Auto-generation modeling for safety-critical software should be suggested using the following procedure.

(1) Environmental modeling as an activity chart

An environmental modeling provides a logical, global view of the steam generator level trip. It defines the environment of the steam generator level trip and the interfaces between the steam generator level trip and the environment. Environmental modeling allows for investigating and specifying the boundaries of the steam generator level trip and determining the inputs and outputs. The environmental modeling procedure is created as follows.

First, the steam generator level trip is named and candidate terminators are identified. Second a draft event response list is made. Third, the event response list for each essential activity is reworked. Finally, a final set of terminators are determined and connected.

(2) Behavioral modeling as statechart

The behavioral modeling describes the detailed functionality. A formally defined set of procedures are as follows. First, detail specifications of the steam generator level trip behavior. Second, describe the data processing(transformation). Third, describe the control logic. Finally, describe data dependencies and intermediate data. Behavioral modeling is derived from the environmental modeling[3].

(3) Implementation modeling as module chart

At the level of the implementation model, the system designer is primarily trying to decide how the environment model and the behavioral model

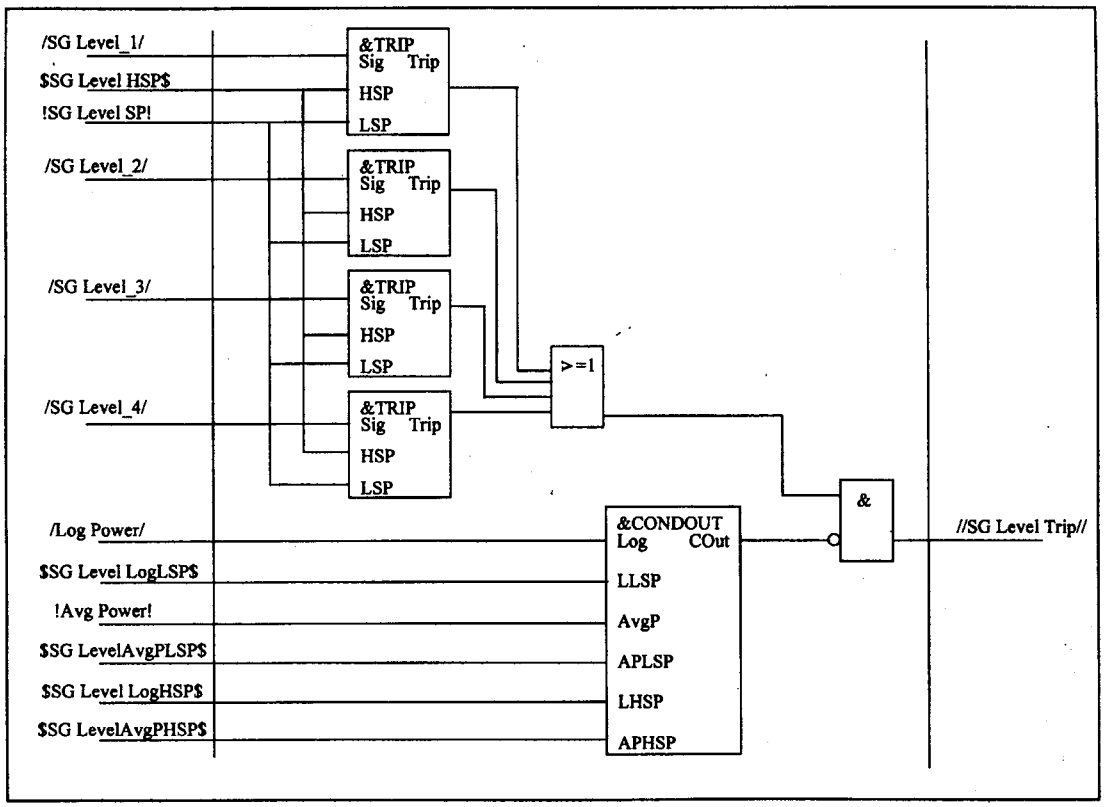

Fig. 1. Trip Logic of Nuclear Reactor Trip by Steam Generator Water Level in Wolsung 2/3/4

should be allocated to different module charts and which module charts should communicate with one another. In this paper, only one activity at a time can take place. This is the special case of the modeling. Steam generator level trip modeling is an optional process.

(4) Auto-generation by Statemate MAGNUM toolset based on a module chart

A module chart or an activity chart(allocated statechart; special case) can be generated by profile on Statemate MAGNUM toolset automatically. Before auto-generation, module charts should be included the graphic panel design. Therefore, the software designer performs only the analysis phase and the design phase in the software life cycle.

2.1. Requirement Analysis of Nuclear Reactor Trip by Steam Generator Water Level

The steam generator water level trip signal is dependent on a) Trip detection using steam generator water level measurements and b) Trip conditioning using reactor power measurements

① It is true when a trip is detected and it is not disabled

② Otherwise it is false

[Trip Detection]

① If any of the four steam generator water level sensors have a value that is greater than or equal to the steam generator water level high setpoint value, the trip detection is true.

② If any of the four steam generator water level sensors have a value that is less than or equal to the steam generator water level low setpoint value, the trip detection is true

③ Otherwise the trip detection is false.

[Trip Conditioning]

① If both the average flux power and ion chamber logarithm power are below their respective low value setpoints then disable trip detection.

② If either of the average flux power or ion chamber logarithm power are greater than or equal to their respective high value setpoints then enable trip detection.

③ Otherwise the trip conditioning remains unchanged.

The logic of a nuclear reactor trip by steam generator water level in Wolsung is shown in Figure 1. It includes trip detection and trip conditioning.

2.2. Environmental Modeling

The purpose of environmental modeling is to produce a context diagram as an activity chart to meet the analysis objectives. It illustrates the analysis scope of nuclear reactor trip parameters by steam generator water level measurements which are the objective of analysis and input/output devices, performance and constraints which the target system should have. The scope of analysis is one of the seven reactor trips of the Wolsong 2/3/4 unit. Input comes from sensors. Output is displayed on the monitor screen and sends out the trip signal when one of the sensors overpasses the limit of the setpoint. In the case where two channel trips(D,E,F channel) occur, the actual trip is displayed and the nuclear reactor trip signal is sent out. When it overpasses the setpoint according to the two thirds logics of D, E, F channel, the nuclear reactor trip occurs. When it overpasses the high setpoint, the nuclear reactor is immediately stopped. Therefore the output device can be defined as a graphic panel screen.

Constraints should be performed in real-time to stop the nuclear reactor safely. The procedure of environmental modeling can be divided into two. The first procedure designs a context diagram. The second procedure names the objectives of analysis and defines input and output devices as peripheral environments. As well, it describes the input/output and body sections for terminators which are the defined input/output devices.

The function in the environmental modeling can be partitioned into two modes; normal and trip, which is further subdivided into four sub-functions. The two modes should be under the control of the control activity which is called condition control(COND_CNTL). The control activity has logpower, LSP(Low Set Point), HSP(High Set Point) and signals of D1, D2, D3, D4, E1, E2, E3, E4, F1, F2, F3, F4. The control activity determines whether the signal values of current input are in the normal scope or in an alarm scope. Normal mode is to classify low alarm, normal, high alarm and display it on monitor screen. If the input signal is in the channel trip range, it is to determine the actual trip mode or not by the decision of the control activity (Figure 2).

2.3. Behavioral Modeling

Behavioral modeling is the next step of a context diagram and is used to analyze the function of lower levels using an activity chart or statechart. In the lower steps, it is used to represent activity for representing function or an activity for

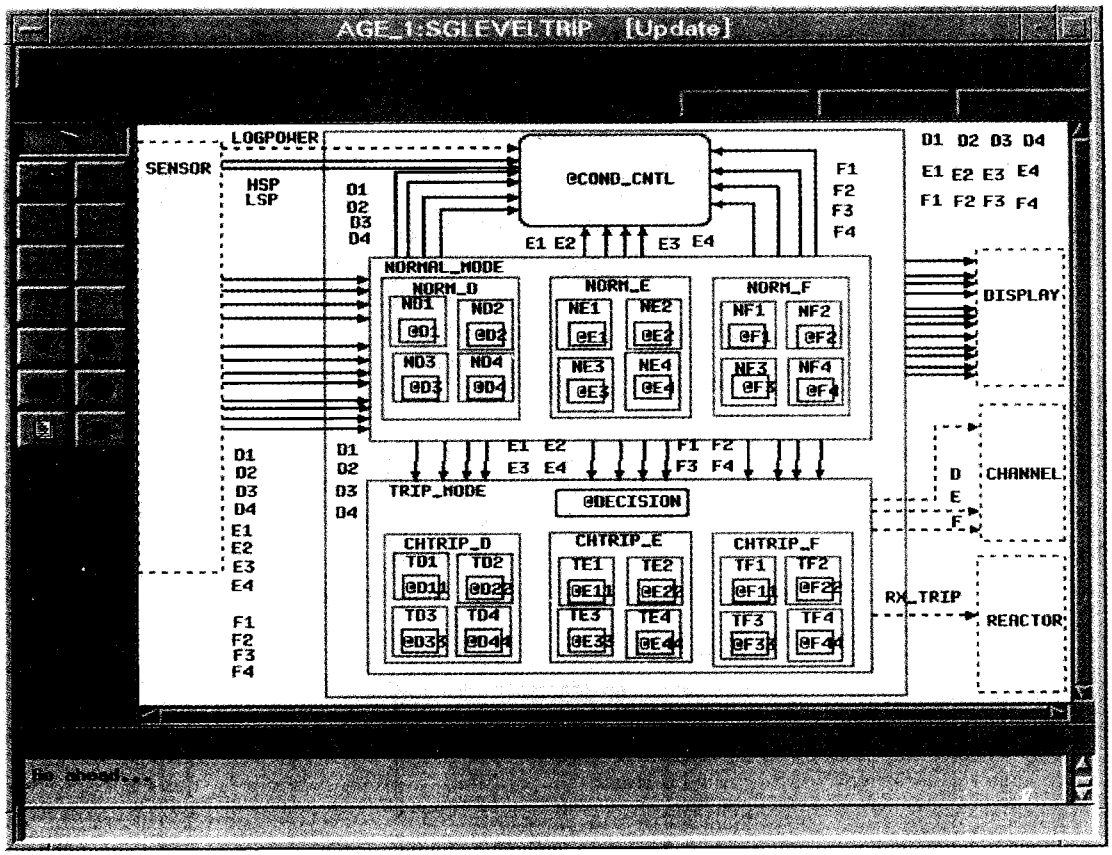

Fig. 2. Environmental Modeling

representing the control as a statechart. In the case of the lower steps of structural hierarchy, it is possible to partition continuous activity chart formats. In example COND_CNTL, if a reading value from the sensor is out of the range of the high or low setpoint, the control is to activate a channel trip mode. If the reading value is in the normal range or in an alarm range, the control is to activate normal mode (Figure 3).

Figure 3 is summarizes the behavioral modeling of control activity.

First, all control activity can be represented as a statechart or P-spec(process specification) type. Second, the higher step' s activity(parent activity) and lower activity(child activity) should be matched harmoniously. Third, data dictionary, for example, event, condition, data-item should be defined until at a primitive level.

2.3.1. NORM_D Behavioral Modeling

The function of NORM_D is to operate under the control of COND_CNTL when the input signal of D1, D2, D3, D4 is in a normal state. If the input signal of D1, D2, D3, D4 is normal, it is to convert the volt of the input signal to a percent of the engineering unit conversion value. When an input signal of D1, D2, D3, D4 is in the alarm range of Lo, it is to convert the input signal value of volt to percent and simultaneously generate a

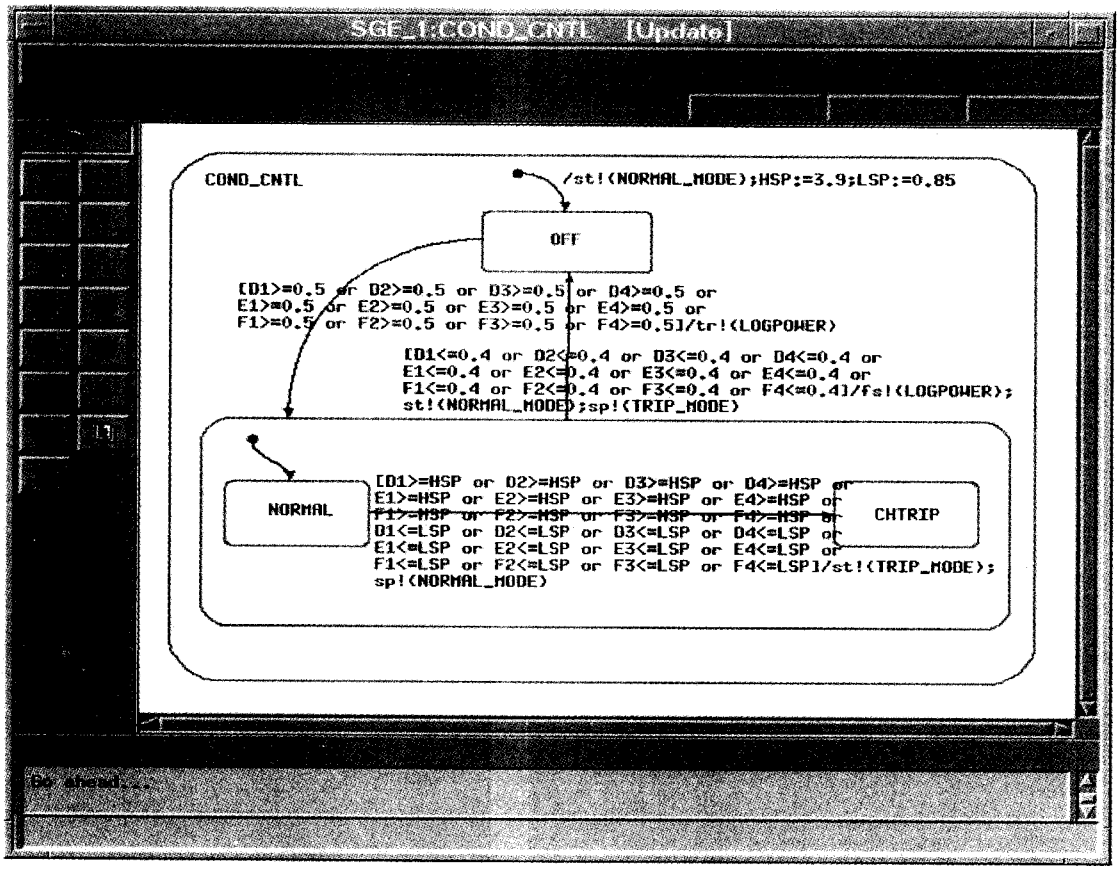

Fig. 3. Behavioral Modeling

Lo alarming event. In the case of NORM_E, NORM_F is the same.

2.3.2. CHTRIP_D Behavioral Modeling

The function of CHTRIP_D is to operate under the control of COND_CNTL when the input signal of D1, D2, D3, D4 is in trip range. If the input signal of D1, D2, D3, D4 is in the trip range due to Hi, it is to convert the volt of input signal to a percent of the engineering unit conversion value and then after trip Hi channel. And if the input signal of D1, D2, D3, D4 is in range of Lo trip, it is to generate Lo channel trip event. In the case of CHTRIP_E and CHTRIP_F, it is the same.

2.3.3. Behavioral Modeling of Decision Control Activity

If only one channel trip is currently occurring, the decision control activity maintains the channel trip unchanged. If more than one channel trip has occurred, it is to trip the nuclear reactor by applying 2 out of 3 logic (Figure 4). At this moment, if the trip is due to more than two Hi channels, it stops the turbine first and then after shutdown, the nuclear reactor. But in the case of a Lo channel trip, it stops the nuclear reactor

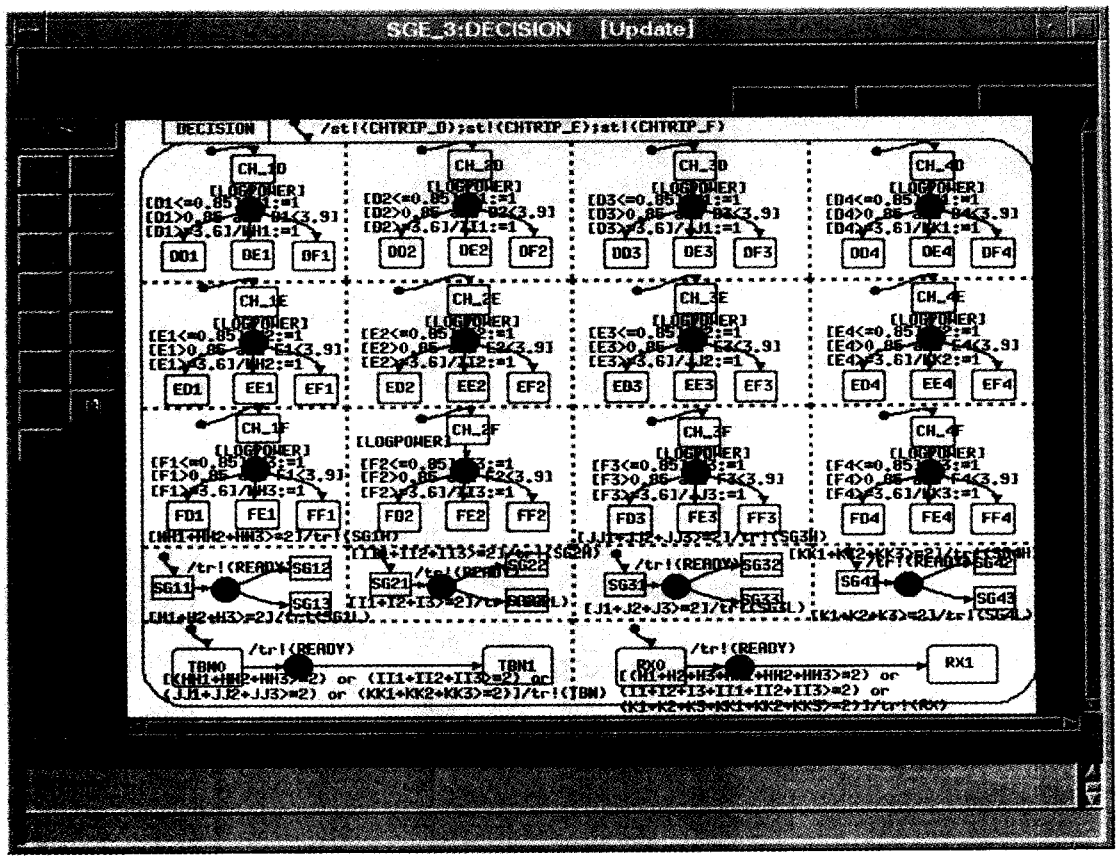

Fig. 4. 2/3 Decision Logic Modeling

immediately.

2.4. Graphic Panel Design

Graphic panel is the graphic user interface. It displays input signals from each sensor. It also displays Lo and Hi alarming messages from an input signal and also displays the trip state of Lo and Hi channels. D, E, F channel trip displays stop messages of the turbine and nuclear reactor on the actual trip windows by the 2 out of 3 logic if the trip comes from a Hi channel. If it comes from a Lo trip, it displays the message for an immediate stop of the nuclear reactor on the actual trip windows(Figure 5). After the graphic panel is completed, it is to bind the graphic panel with the modeling of a context diagram and behavioral modeling. It then defines attributes of Data Dictionary Entry(DDE) and performs the verification and validation processes.

2.5. Module Chart Design

The module Chart Design allocates each function modeled by the analysis steps to each module by applying the software engineering principles for example, boss rule, transform rule and transaction rule by the top down approach. In the case of nuclear reactor trip parameters in general, it is composed independently due to safety problems.

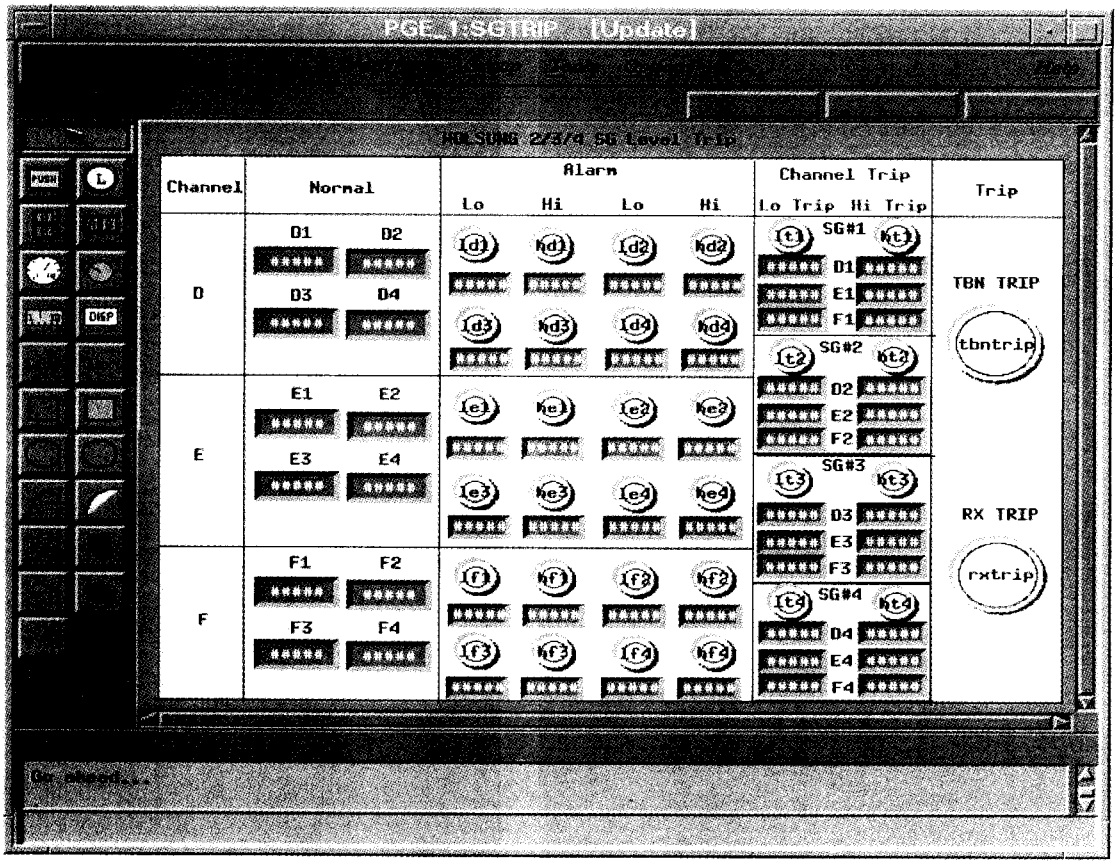

Fig. 5. Graphic Panel Design

Therefore in the case of Wolsong 2/3/4 unit reactor trip parameters, a total of seven trip parameters have one-to-one correspondence modules. When one function allocates one module, it can design a module chart since it is a very special case. The reason why is that it is to modulize by allocating the analyzed modeling to the design model one-to-one correspondence in the process of automatic code programming. In these cases, the software complexity of objective software is low and simple, which is a very special case. When software complexity is low, it can be a one-to-one correspondence of function to module, which is the exact same case. If it is not 1:1 correspondence (that is M:1 or M:N), it should design a module chart.

2.6. C Language Auto-generation and Execution

After analysis and design is completed, it is to produce C language automatically using Sharpshooter/C code generator which is part of the Statemate MAGNUM toolset in CASE tools (Figure 6)[1][2]. If it needs external programs for interface before C language is produced, it is to produce code automatically by inserting a test bench program, communication program or data

```
        setc(&SG3L,TRUE);
        notify(scope_id,conSG33,TRUE);
        STATE_82_isin = SG33;
      }
    }
} /* exec_STATE_82 */
void exec_STATE_81()
{
    if(STATE_81_isin == SG21 ) {
      if ((II1 + II2) + II3>=2) {
        notify(scope_id,conSG21,FALSE);
        setc(&READYco13,TRUE);
        setc(&SG2H,TRUE);
        notify(scope_id,conSG22,TRUE);
        STATE_81_isin = SG22;
      }
      else if ((I1 + I2) + I3>=2) {
        notify(scope_id,conSG21,FALSE);
        setc(&READYco13,TRUE);
        setc(&SG2L,TRUE);
        notify(scope_id,conSG23,TRUE);
        STATE_81_isin = SG23;
      }
    }
} /* exec_STATE_81 */
void exec_STATE_80()
{
    if(STATE_80_isin == SG11 ) {
      if ((HH1 + HH2) + HH3>=2) {
        notify(scope_id,conSG11,FALSE);
        setc(&READYco13,TRUE);
        setc(&SG1H,TRUE);
        notify(scope_id,conSG12,TRUE);
        STATE_80_isin = SG12;
      }
      else if ((H1 + H2) + H3>=2) {
        notify(scope_id,conSG11,FALSE);
        setc(&READYco13,TRUE);
```

Fig. 6. Example of C Language Auto-Generation

acquisition program. When C language is generated automatically by the code generator, an execution file can be made by compiling and linking procedures. The source code and execution code which are a final product, can be ported to another machine independently and can be modified and recompiled. When it needs a little modification for the source code, it is to modify and recompile the source code itself instead of repeating the procedures of the first modeling step and C language automatic programming steps.

3. Conclusions

A strategy and a procedure to model the auto-generation of safety-critical software using Statemate MAGNUM toolsets have been suggested. Methods for environment modeling and behavioral modeling have also been recommended. The modeling strategy describes a set of modeling processes for safety-critical software development of a well-defined steam generator level trip in Wolsong 2/3/4.

This thesis is to model the trip logic caused by the steam generator water level of the Wolsong Nuclear Power Plant 2/3/4 unit and generate C language automatically. Automatically generated C language can be verified and validated and be produced with less effort and less human power and with higher quality. Also it can be certain to be portable and machine independent since it is generated with the C language.

The resulting product by Statemate MAGNUM toolset may be demonstrated to be of sufficient quality for an intended application domain in a safety-critical software development in the nuclear field where any Verification and Validation (V&V) methodologies exists.

References

1. *STATEMATE MAGNUM Reference Manual*, i-Logix, (1997).
2. *STATEMATE MAGNUM Trailblazer Reference Manual*, i-Logix, (1997).
3. *Modeling Reactive System with Statechart Approach*, DAVID HAREL AND MICHAL POLITI, (1997).
4. ERICSON, C,A. 1981. *Software and System Safety. In Proceedings of the 5th International System Safety Conference* (Denver, Colo.), Vol/ 1, part1. System Safety Soc., Newport Beach, Calif., pp. III-B-1 to III-B-11.
5. ROLAND, H.E., AND MORIARTY, B. 1983. *System Safety Engineering and Management.* Wiley, NewYork.
6. WELLBOURNE, D. 1974. *Computers for Reactor Safety Systems.* Nucl. Eng. Int.(Nov.), 945-950.