ICOT Technical Memorandum: TM-0406

TM-0406

Support System for Specification Phase of Communication Software

by

K. Shibata, W. Tanaka,
& H. Hasegawa

November, 1987

ICOT

Mita Kokusai Bldg. 21F
4-28 Mita 1-Chome
Minato-ku Tokyo 108 Japan

(03) 456-3191~5
Telex ICOT J32964

Institute for New Generation Computer Technology

SUPPORT SYSTEM FOR SPECIFICATION PHASE OF COMMUNICATION SOFTWARE

Kenji SHIBATA, Wataru TANAKA and Haruo HASEGAWA

Telecommunication Division
Oki Electric Industry Co., Ltd.

ABSTRACT

We have been developing the software support system -- EXPRESS-- which designs the specification for a communication system. Specification phase is the most important in developing communication software. Expert engineers spend a lot of man-hours to upgrade a communication system. EXPRESS has been built up according to the design procedure used by experts. EXPRESS designs the specification according to users' requirements written in a natural language and verifies the specification. Each requirement is converted into PSG(Partial Service Graph) and all PSGs are integrated into TSG(Total Service Graph). TSG is the final design specification. PSG and TSG are described by using petri nets. PSG and TSG may have ambiguities and inconsistencies which are included in the original requirements or made in the process of integrating PSGs into TSG. These ambiguities and inconsistencies should be detected and resolved. The verification of the specification consists of syntactic verification and semantic one.

1. INTRODUCTION

In recent years, the service features have become various and advanced in a communication system because of sophisticated networks. And it takes a lot of man-hours to develop communication software. It is important to design the specification from users' requirements accurately because the specification is a fundamental document of the next phases in communication software. In order to settle the problem, we have been developing the support system EXPRESS (EXPeRt system for ESS) which automatically designs the specification for a communication system. EXPRESS understands users' requirements written in a natural language, designs the specification and verifies it.

First, this paper describes the design procedure used by experts, clarifies required conditions of the support system and outlines the configuration of the system. Then it describes how EXPRESS designs and verifies the specification.

2. DESIGN

The following is the process in which experts design the specification accurately from users' requirements:

(1) Listen to users' requirements
  Users' requirements are fragmentary in general.

(2) Clarify the service specification for each requirement.
  Compare each of fragmentary users' requirements with known services and choose some services which seem to meet the requirements. Show the operation of terminals in the services and verify whether to meet the users' requirements or not. For a new service, clarify the operation on the basis of fragmentary requirements. If the operating method is inconsistent with the hardware function, give up the service or propose an alternative service.

(3) Integrate the above service specifications.
  Unify the same states in the above service specifications. Resolve inconsistencies that though two of the above specifications have the same present states and the same operations, the state of one service specification is different from that of the other after the operation. If it is impossible to resolve the inconsistency, either give up the service or propose an alternative service. Check whether new services are included in the integrated specification or not. If new services exist, ask to users if the services are allowed.

(4) Add the error processing to the specification.

The support system must be built up according to the design procedure. Required conditions for the support system are as follows:

(a) Easy to input requirements
(b) Easy to verify the specification
(c) Divided into two kinds of stages : one is a partially designing stage and the other is a totally designing stage
(d) Knowledge for verification
(e) Stored with the specification as knowledge
(f) Verifiable whether the specification meet users' requirements or not

EXPRESS is the support system which satisfies the above conditions.

3. SYSTEM CONFIGURATION

Fig.1 shows a system configuration of EXPRESS. The system consists of requirement understanding subsystem, specification integrating subsystem, knowledge base and specification verifying subsystem.
The functions of them are as follows:

(1) Requirement understanding subsystem

When users input an ambiguous requirement about service of a switching system in a natural language, this subsystem understands the requirement by utilizing the knowledge in the knowledge base. This subsystem transforms each requirement into an individual

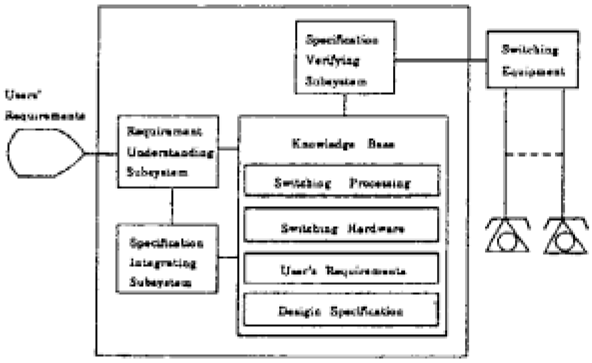

Fig.1 System Configuration of EXPRESS

specification. When the users' requirement is inconsistent with the knowledge base, the inconsistency is resolved interactively.

(2) Specification integrating subsystem

The individual specifications is integrated into a final specification by this subsystem. Each of individual specifications is fragmentary and they may be inconsistent with each other. When some inconsistencies are made in this process, they are also resolved interactively.

(3) Knowledge base

The specification which was designed and the knowledge required to design the specification are stored in the knowledge base. The knowledge is related to a switching processing, switching hardware, users' requirements, etc.

(4) Specification verifying subsystem

This subsystem controls a switching equipment according to the specification stored in the knowledge base and verifies whether the specification satisfies semantically users' requirements or not.

4. SPECIFICATION DESIGN

4.1 Specification expression

Fig.2 shows the process of designing the specification without inconsistency or ambiguity.

The process in which experts design the specification from the users' requirements can be considered as the conversion from the service expression representing the users' requirements into that representing the specification. Each service expression is explained below:

(1) SE(Service Element)

Each of users' requirements is converted into SE at first. SE describes the terminal operation and the terminal state changed by the operation from the beginning to the end of the service. Fig.3 shows an example of SE.

```
se(knt(kn(station-to-station_call,
          caller_first_goes_on-hook),time(1)),
   ([tr(off-hook,[agent(caller)])],
    [st(hear,[goal(caller),object(dial_tone)])])).
```

Fig.3 An example of SE

(2) PSG(Partial Service Graph)

SE is converted to PSG which is described in petri nets. In PSG, an action is expressed as a transition, a state as a place and a terminal as a token. Each PSG consists of a present state, an action and a next state. A present state is represented as a set of input places and a next state as a set of output places. Fig.4 shows an example of PSG.

```
psg(knt(kn(station-to-station_call,
           caller_first_goes_on-hook),time(1)),
    spp([place([rel(token(caller,ext),idle,nil)]),
         place([rel(token(dial_tone,dt),idle,nil)])]),
    tr(token(caller,ext),off-hook,nil),
    snp([place([rel(token(caller,ext),
                receive ,token(dial_tone,dt))])])).
```

Fig.4 An example of PSG

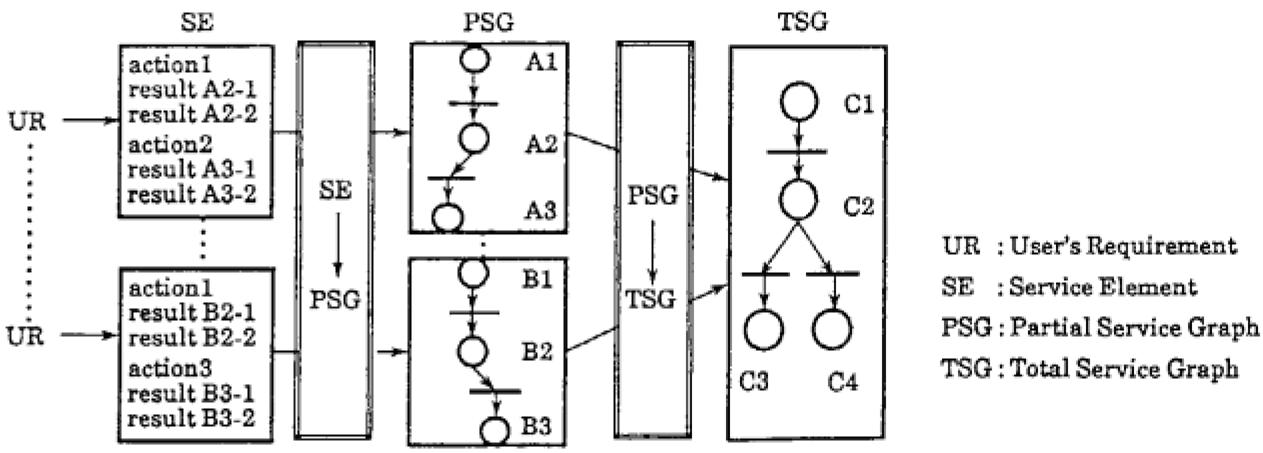

Fig.2 The process of the designing the specification

(3) TSG(Total Service Graph)

TSG is the final specification, which is designed by integrating several fragmentary PSGs. A single state corresponds to a place and an action to a transition. TSG is expressed as a set of these places and transitions, and the connection between a place and a transition is clarified. Fig.5 shows an example of TSG.

```
place(Pt1,gpl([grel(gtoken(Ttk1,ext),idle)]),
          spt([Tt2,Tt4,Tt7,Tt8]),
          snt([Tt1,Tt3,Tt5])).
transition(Tt1,Arc1,off-hook,transit([Arc1,Arc2]),
               spp([Pt1,Pt6]),
               snp([Pt2])).
arc(Arc1,in(Ttk1),out(Ttk3),atr([])).
```

Fig.5 An example of TSG

As mentioned above, both PSGs and TSG correspond to petri nets. SDL (Specification and Description Language) recommended in CCITT is well known among the languages describing the specification in a communication system. Petri nets are used in order to model and analyze dynamic systems, and the feature of petri nets is to simplify verification and analysis. In addition, EXPRESS uses obtained specification as knowledge. In this case, petri nets are considerd to be effective for specification representation because the specification can be easily added or modified by discribing it in petri nets.

4.2 Example of designing specification

The integration of PSGs into TSG, resolution of inconsistencies produced during integration and creation of new services are described here. (a),(b) and (c) in Fig.8 represent petri nets of PSG obtained by users' requirements shown as (a), (b) and (c) in Fig.6 respectively. No error exists in these PSGs.

(1) Integration

PSGs are integrated into TSG by searching and unifying the same states and the same actions of PSGs. The state is expressed as the relation between terminals. The same state means that the types of terminals are the same and the relations are the same between terminals. The same action means that the types of action are the same and the states of the terminals before the action are the same.

(2) Inconsistency Resolution

The integration shown as (a) and (b) in Fig. 8 is made by the above method. However, there is inconsistency between (a) and (c). In the portions P11→T11→P12 of (a) and P31→T31→P33 of (c) in Fig.8, next states of them differ from each other though they have the same present states and the same action. In this case, two telephone sets should be separately handled. For example, the hot line attribute is given to the telephone sets of (c). As a result the integration can be made.

(3) Creation of new services

It may be possible that features other than those of requirements are included in TSG. Whether to allow these services to be created or not should depend on user's will. For the above case, the requirements correspond to P1→T5→P3→T4→P1 and P2→T3→P3→T6→P4 as shown in Fig.7 and the following two new routes are obtained from TSG : P1→T5→P3→T6→P4 and P2→T3→P3→T4→P1.

1. A caller goes off-hook.
   Then the caller hears a dial tone.
2. The caller dials the number of called party.
   Then the caller hears a ringback tone.
   A called party is rung.
3. The called party goes off-hook.
   Then the caller and the called party talk to each other.
4. The caller goes on-hook.
   Then the called party hears a busy tone.
5. The called party goes on-hook.
   The busy tone stops.

(a) station-to-station call, the caller first goes on-hook

1. A caller goes off-hook.
   Then the caller hears a dial tone.
2. The caller goes on-hook.
   The dial tone stops.

(b) station-to-station call, the caller goes on-hook while hearing a dial tone

1. A caller goes off-hook.
   Then the caller hears a ringback tone.
   A called party is rung.
2. The caller goes on-hook.
   Then the ringback tone and the ringing signal stop.

(c) hotline, the caller goes on-hook while hearing a ringback tone

Fig.6 Users' Requirement

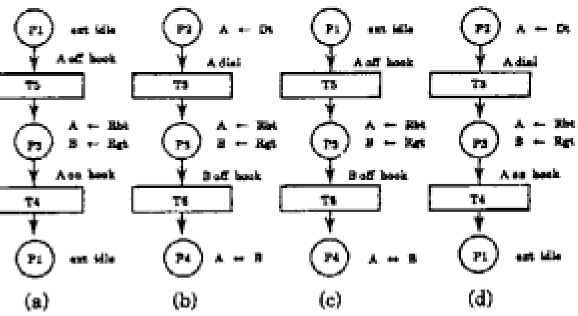

Fig.7 Representation of a new service

5. SPECIFICATION VERIFICATION

Users' requirements may be frequently modified and added in a switching system. There is a possibility that the specification includes some inconsistencies, because users' requirements have already included some inconsistencies or they can be made in the process of designing the specification. Therefore it is necessary to detect and resolve inconsistencies during the design of the specification. It is also important to verify whether the specification satisfies requirements or not.

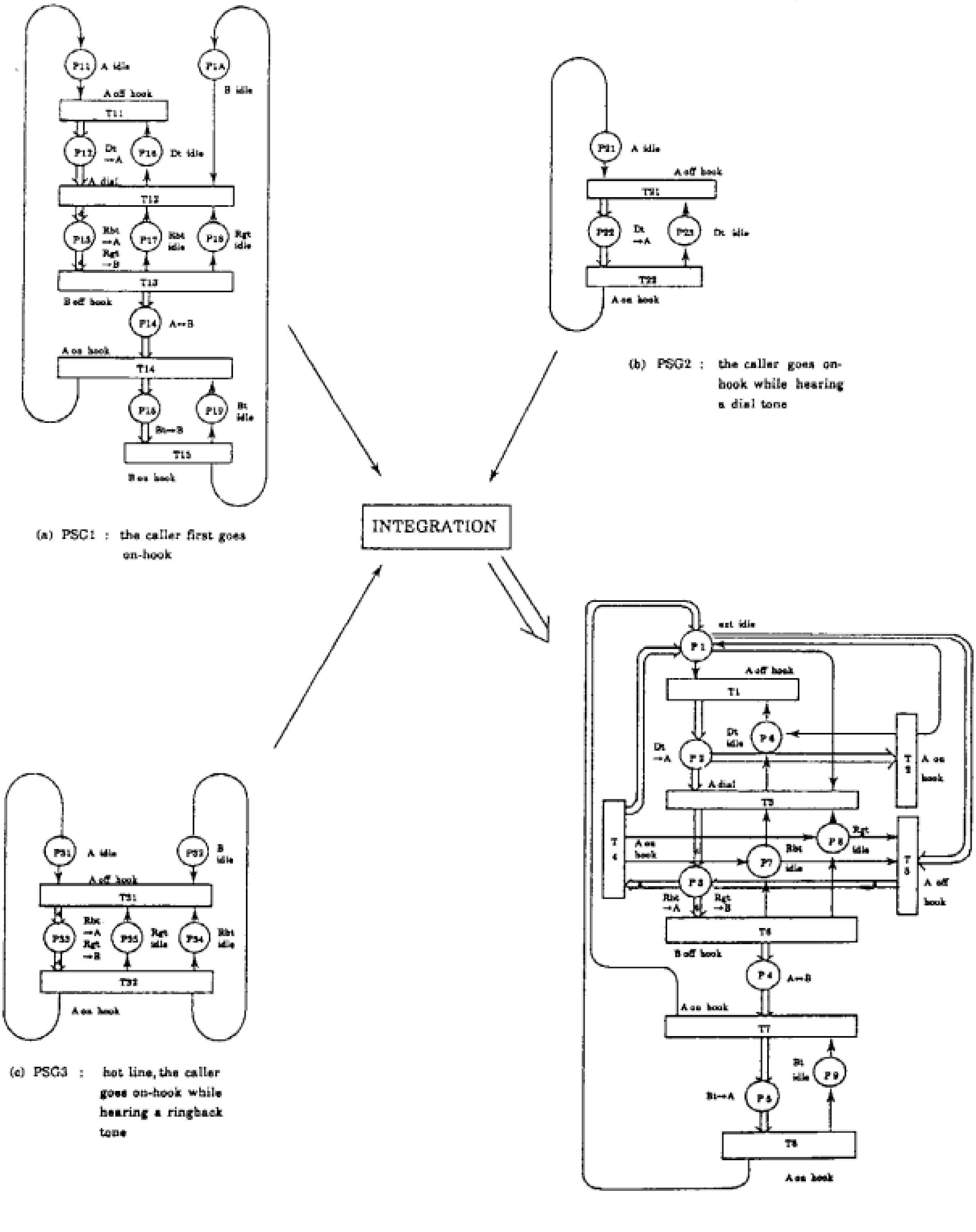

Fig. 8 Representation of PSGs and TSG in Petri Net

EXPRESS executes the following verification:

(1) Verify whether the specification is syntactically correct
(2) Verify whether the specification is semantically correct

5.1 Syntactic verification

Syntactically correct specification means that PSGs and TSG described in petri nets are syntactically correct. The syntactical inconsistency consists of two types : one is included in each PSG and the other is produced when PSGs are integrated into TSG. For example, there is a case in which the state dose not return to the initial state after tokens have passed through all or several transitions. The syntactical inconsistency like the above can be detected by the analysis method of using petri nets matrices.

5.2 Semantic verification

Semantically correct specification satisfies the following properties:

(1) uniqueness: the specification is unique and has no ambiguity
(2) consistency: the specification has no inconsistency
(3) requirement satisfactory: the specification satisfies users' requirements completely.

Users do not always have accurate knowledge about a communication system. Therefore they may use ambiguous words or require an operation which cannot be realized. Knowledge about switching is used for designing the specification in order to resolve ambiguity and inconsistency. In addition, EXPRESS works a switching equipment to verify the specification. Users can confirm their requirements by operating terminals actually.

6. CONCLUSIONS

This paper has clarified the procedure used by experts designing a communication system and outlined the specification design support system which was made according to the procedure. The program language of the system is ESP(Extended Self-contained Prolog) on PSI(Personal Sequential Inference machine). At present, EXPRESS designs the specification for certain services from users' requirements and controls the switching equipment according to TSG. It is confirmed that the users can themselves perform the semantic verification about the specification. It is also confirmed that petri nets are applicable to the expression of the knowledge for services.

ACKNOWLEDGMENTS

The authors thank all the members of the EXPRESS project for useful discussions. We would like to be thankful to Mr. R. Hasegawa of ICOT(Institute for New Generation Computer Technology) for many helpful suggestions.

REFERENCES

[1] W.Tanaka and H.Hasegawa : "Specification Design for a Communication System by a Prototyping Method", SIG Software Engineering of IPS Japan, 87-SW-55-3 (1987) [in Japanese]

[2] H.Hasegawa, W.Tanaka and K.Shibata : "Analysis of Design Specification in a Communication System by means of Petri Nets", COMPSAC (1987)

[3] J.L.Peterson : "Petri Net theory and the Modeling of Systems", Prentice-Hall (1981)

[4] CCITT Recommendation : "Functional Specification and Description Language (SDL)", CCITT REDBOOK (1984)

MAILING ADDRESS

Telecommnuication Division
Oki Electric Industry Co.,Ltd.
4-11-22, Shibaura, Minato-ku, Tokyo 108, JAPAN