# 30

# Application of Protocol Synthesis Technique to Resolution of the Service Interaction Problem

Yoshiaki KAKUDA†, Hiroyuki ASADA† and Tohru KIKUNO†

†Department of Information and Computer Sciences, Faculty of Engineering Science, Osaka University, 1-3 Machikaneyama-cho, Toyonaka-shi, Osaka 560, Japan

This paper proposes a method for resolving the service interaction problem in the Advanced Intelligent Network [1] [5]. Since this method is based on the protocol synthesis technique, the service interaction problem is resolved in the protocol specification rather than in the service specification. As a result, invalid interactions between communication services can be dynamically avoided.

## 1. Service Interactions and Specifications

### 1.1. Service Rules

Since a communication system consists of multiple users, a state of the communication system can be composed by a set of states of such users in the system. In the STR description method [2], a state of each user is represented by a collection of predicates. A service specification is described by service rules, each of which consists of (1) current states of users, (2) an event triggered by a user, and (3) next states of users. Two examples of service rules are shown in Figure 1.

### 1.2. Service Interactions

For a given state of a communication system (simply, system state) and service rules, if all predicates in current states of users are satisfied by the system state, then the event is said to be executable in the system state. If there exist two service rules including the same executable event in the given system state, then it is said that a service interaction occurs for these two service rules in the system state. If the next states in two service rules for which the service interaction occurs contradict with each other, then the service interaction is invalid and one of them should be chosen.

| | |
|---|---|
| path(A,B), out-dialtone(C), m-cw(A) | ··· current state |
| dial(C,A): | ··· event |
| out-ringback(C,A), cw-ringing(A,C), path(A,B), m-cw(A) | ··· next state |

(a) Call Waiting (CW) Service Rule

| | |
|---|---|
| out-dialtone(C), path(A,B), idle(D), m-cfv(A,D) | ··· current state |
| dial(C,A): | ··· event |
| out-ringing(D,C), out-ringback(C,D), path(A,B), m-cfv(A,D) | ··· next state |

(b) Call Forwarding (CFV) Service Rule

Figure 1. Examples of Service Rules

D. Hogrefe et al. (eds.), *Formal Description Techniques VII*

Suppose that seven predicates m-cw(A), m-cfv(A,D), path(A,B), path (B,A), out-dialtone(C), idle(D) are satisfied by a certain system state. Then, a service interaction occurs for the CW and CFV service rules in Figure 1 because these two rules include the same event dial(C,A) that is executable in the system state. Since out-ringback(C,A) in the next state of the CW service rule and out-ringback(C,D) in the next state of the CFV service rule contradict each other with single path connection, one of two dial(C,A)s should be chosen.

### 1.3. Protocol Specification

Service primitives, shortly primitives, are transmitted and received between users in an upper layer and processes in a lower layer through an interface called Service Access Point(SAP). The confirmation, indication, request and response (which are abbreviated as conf, ind, req and resp, respectively) are examples of primitives. A typical example of primitive sequence is req, ind, resp and conf. Messages for synchronization are transmitted and received between processes. A protocol specification is modeled by finite state machines(FSMs), which include sequences of primitives and messages.

## 2. Resolution of Service Interactions

In this paper, the service interaction problem is defined as follows:

**Input** ··· A service specification described by STR description method.

**Output** ··· A protocol specification with such capabilities that resolve the service interaction. For ease of understanding, the protocol specification is represented by sequence charts, which are equivalent to the FSMs.

The proposed method for resolution of the service interaction problem is as follows. First, for each communication service in a given service specification, a sequence of primitives is generated from the service rule. Next, a protocol specification that may include service interactions is derived from all the generated sequences of primitives. This derivation is done by the protocol synthesis technique [4] [6]. Finally, the service interaction is resolved by inserting conditional branch functions to the protocol specification.

As explained in Section 1, the service interaction can be interpreted as a non-deterministic behavior of executable events in the service specification. Since each event in the service specification is translated into a sequence of primitives in the protocol specification, the service interaction is regarded as a non-deterministic execution of primitive sequences. The first half of one primitive sequence includes that of the other, but the latter half is different with each other. Thus the service interaction problem can be basically resolved by inserting, to the protocol specification, conditional branch functions for determining which primitive sequences are selected. The details of this resolution method are shown in [3].

Figure 2 shows a sequence chart representing the protocol specification derived from CW and CFV services which cause service interaction. In this figure, black thick lines represent the sequence in the case that CW service is chosen, and gray thick lines represent the sequence in the case of CFV service.

## 3. Conclusion

This paper has proposed a method for resolution of the invalid service interaction problem, which is based on the protocol synthesis technique. In this method, non-deterministic

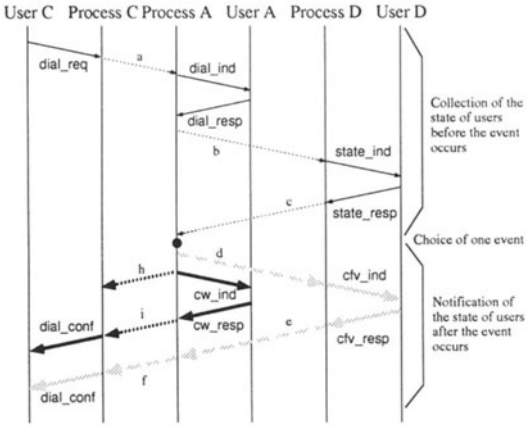


Figure 2. Sequence Chart Representing Protocol Specification Generated by Protocol Synthesis

events in service specification are transformed into deterministic sequences of service primitives with conditional branch functions in the protocol specification. In case of valid service interactions, multiple non-deterministic events only have to be executed simultaneously.

## REFERENCES


1. N.Griffeth and Y.-J.Lin(eds.), Special issue on managing feature interactions in telecommunications systems, IEEE COMPUTER, Vol.26, No.8(Aug.1993).
2. Y.Hirakawa and T.Takenaka,"Tecommunication service description using state transition rules," Proc. Sixth Int'l Workshop on Software Specification and Design, pp.140-147(Oct.1991).
3. Y.Kakuda, H.Asada and T.Kikuno,"Protocol synthesis technique and its application to the feature interaction problem," Proc. ATR International Workshop on Communications Software Engineering(Oct.1994) to appear.
4. Y.Kakuda, M.Nakamura and T.Kikuno,"Automated synthesis of protocol specifications from service specifications with parallelly executable multiple primitives," IEICE Trans. on Fundamentals of Electronics, Communications and Computer Sciences(Oct.1994) to appear.
5. Y.-J.Lin and N.Griffeth(eds.), Special issue on managing feature interactions in telecommunications systems, IEEE COMMUNICATIONS MAGAZINE, Vol.31, No.8(Aug.1993).
6. K.Saleh,"Automatic synthesis of protocol specifications from service specifications," Proc. Int'l. Phoenix Conf. on Computers and Communications, pp.615-621(Mar.1991).