An Algorithm For NGN Feature Interaction Detection

Yiqin Lu\(^1\), Guangxue Yue\(^2\), and Jiajin Wang\(^1\)

\(^1\)School of Electronic and Information Engineering, South China University of Technology, Guangzhou, China
eeyglu@scut.edu.cn

\(^2\)Jiaxing University, Jiaxing, China,
guangxueyue@163.com

Abstract—With service design open to third-party in the evolving telecom networks, dynamic method is suitable for feature interaction (FI) detection. When implementing this method, however, the violation of user intension is difficult to express as an indication of FI. To overcome this, a duplicate of Application Server called FRS (Feature-Representation Server) is proposed to add into the network, where each feature can run again individually and the running results are sent back as desirable results to compare with those obtained in the working environment. Interaction exists if they are inconsistent. An algorithm for detecting such kind of FI is given with an experiment for demonstration.

Index Terms—feature interaction, NGN, algorithm, detecting

I. INTRODUCTION

The emerging telecommunication networks are service-driven networks, i.e., features will be designed and integrated into the networks in an easier way, and number of service will increase very fast. Here ‘feature’ is the smallest part of a service provided by the system [1], such as Call Waiting (CW), Call Forwarding (CF), Number Portability (NP), Originating Call Screening (OCS) etc., and FI is such a phenomenon that two or more activated features interfere with one another and cause abnormality of the system [1-7]. For example, a restaurant with several branches subscribes the feature ‘One Number (ONE)’ for delivery service. The feature may connect an incoming call to the nearest branch according to the caller’s number. Suppose a man living in district A calls the number of delivery service, his call is connected to branch \( d_1 \) of the restaurant. One day, he removes to district B, where the nearest branch is \( N_B \), but he subscribes the feature NP to reserve his telephone number used in district A. Now, if he calls the number of delivery service, his call is still connected to branch \( N_B \) instead of \( N_B \).

Another example of FI is the interaction between CF and OCS. CF forwards all incoming calls to another phone. OCS prohibits calling to certain callee. When CF and OCS are both activated in the network, FI may occur as follows: Consider three phones \( d_1 \), \( d_2 \) and \( d_3 \). Suppose \( d_1 \) has activated feature OCS forbidding a call to \( d_2 \) and \( d_3 \) has activated feature CF to \( d_2 \). When \( d_1 \) calls \( d_3 \), the call is forwarded to \( d_2 \), so that \( d_1 \) can still connect to \( d_2 \), which OCS does not expect.

FI existed when the telecommunication system was in PSTN (public switch telephone system) age, and it become more serious when the telecom system evolves to a service-driven network such as IN (intelligent network) and NGN (next generation network), because the number of features increases very fast in a service-driven network.

FI has been studied under three classes of problems: detection [8], avoidance [9] and resolution [10]. Among these classes, detection is investigated the most, because it is the inevitable step in the management of FIs.

FI detection can be carried through in an on-line or off-line manner [3,7,11]. On-line detection is also called static detection. It often uses formal methods such as SDL, FSM, LOTOS, Petri nets [3,12-16] to specify features and verify the system correctness with some criteria. Off-line detection is called dynamic detection. It is realized by capturing the run-time behaviors of the features and comparing with the stored ‘correct’ behaviors, which could be learned in a test environment [3,7,17-19].

Since the static detection of FI is easier to realize and can find the problem in an earlier stage of a feature lifecycle, most existing literatures are based on this manner [6,12-16]. However, with the telecom networks evolving into NGN, static detection is faced with challenge: in a NGN service-supporting environment, feature design is open to the third-party developers. One can develop a feature using the programming language (like C++ or Java) without getting into the details of network and signaling. For business purpose, the feature logic will be kept private. This makes it hard to specify a feature with formal languages. Therefore, dynamic detection is preferred to use to detect FI in NGN [4,5,7].

Roughly, the FI can be divided into two kinds [17]:

- Technical interaction: two features run simultaneously and lead the system into inconsistent states or to exhibit inconsistent observable actions.
- Violation of user intension: one or two features fail to meet the user requirements when they run concurrently.
As far as dynamic detection is concerned, the ‘technical interaction’ occurs when one feature disarrange the procedure of another feature and causes it execute abnormally, and the ‘violation of user intention’ occurs when one feature modifies the data or parameter values of another feature and cause it work without desirable results.

For the technical interaction, detection may not be such an arduous task, because there are some structural characteristics in the run-time message sequences when interaction occurs, such as loop, abnormal terminal etc. However, for the intention-violation FI, detection is difficult because it is hard to express the ‘undesirable results’ with the trivial knowledge learned from individual execution instances of a feature.

This paper proposes a dynamic method for detecting intention-violation FI in a NGN environment. In this method, a redundant application services (AS), named Feature-Representing Server (FRS) in the paper, is added into the network. When a feature is created in AS, a copy of the feature is made in FRS. When two (or more) features are activated together, the trigger conditions of each feature are sent to the FRS. Thus each feature can run individually in the FRS and the result are sent back as desirable result to compare with those result got in the working environment. FI occurs when the tow results are inconsistent.

In the rest of this paper, Section II introduces the service creation in NGN and the challenge of FI detection in it. Section III describes the method: firstly, the additional functional entities of FRS and FIM are introduced, then the format of SS-FIM message is described, finally the algorithm of FI detection is presented. Section IV shows the result of implementation of the algorithm. Section V draws a conclusion.

II. SERVICE CREATION IN IN NGN AND THE CHALLENGE OF FI DETECTION

The NGN network is based on a 4-layer architecture, namely Access, Transport, Control, Service. In access layer, the existing telecom networks or user terminals can be connected to a NGN through every kinds of gateway, such as signaling gateway (SG), trunk gateway (TG), network access server (NAS), and access gateway (AG), etc. The core transport network is an IP network. The control of a call or a session is managed by softswitches (SS) in the control layer. The features are running in application servers (AS) in the service layer. Besides the AS, there may be many other servers in the service layer, such as AAA server, network management server (NMS), etc. In such architecture, transport, control and service are separated from each other. That makes the service can be designed by the third-party, who may not know much about the technical details of a telecom network. The AS may provide an API interface to the service designers so that they can simply design a service with ordinary programming language, such as Java, C++, XML, etc. When a service is designed and created, the feature logic will be kept private for business purpose, which makes it hard to specify a feature with formal languages. Therefore, the traditional static FI detection is not suitable for such a framework. In Section III, a dynamic method for FI detection is present, which is focus on the FI cause by ‘user intention violation’ (the second type of FI stated in Section I).

III. INTENTION-VIOLATION FI DETECTION BASED ON FEATURE REPRESENTATION

In this section, a dynamic method for detection of intention-violation FI is presented. As shown in Fig. 1, tow entities, i.e. FIM (feature interaction manager) [18] and FRS (feature-representing server), are added into a NGN, and a FIM interface is embedded into SS.

A. FIM and FRS

FIM communicates with SS through ‘FIM interface’, which captures the SIP flows between ASs and user agents (UA), translates them into so-called ‘SS-FIM message’ and send it to FIM. FRS is a special AS, which keeps a copy of program of every service when it is created in AS. When two (or more) features are activated together, the trigger conditions of each feature are sent to the FRS. Each feature runs in the FRS individually and the running result is sent back to FIM. FIM uses the result as desirable result of the feature and compare with those getting in the working environment. There may be a FI if the tow results are inconsistent.

There may be more than one AS but only one FRS in the network. In such case, every feature in different ASs must create its copy in the unique FRS.

B. SS-FIM message

The format of a SS-FIM message likes a simplified version of SIP. Only the key fields relating to FI are remained. They are Start-Line, Via Header Field, From Header Field, To Header Field, and Call-ID Header Field. Here, Via Header Field plays an important role. For SS-FIM messages flow from SS to FIM is the hybrid of the signaling of all running features. The Branch ID parameter in the Via Header Field values serves as a
transaction identifier to distinguish which feature the message belongs to.

A call is identified uniquely by ‘Call-ID’. Besides, the FRS-FIM message from FRS to FIM has an additional field representing the results of the feature running individually in FRS.

C. Algorithm for FI Detection

In this sub-section, an algorithm is proposed for detecting intention-violation interaction between two features in NGN. Obviously, how to judge the violation of user intention is the first issue. The algorithm addresses this issue as follows (Fig. 2): once a feature is created in AS, a copy will be sent to FRS. When a feature is triggered in SS, the trigger condition is sent to FRS so that the feature is activated in FRS. The feature runs independently in FRS and the running result is sent back to FIM as the desirable result. At the same time, the feature runs in AS and the actual results can be compare with those sent from FRS. If they are inconsistent, the intention is violated.

![Algorithm Diagram]

Figure 2. Judge the violation of user intention.

According the intention of which feature is violated, FI may occur in the following tow cases:

1) The intention of second feature is violated, such as the interaction between NP and ONE. This kind of FI occurs when the first feature changes the initial parameters of the second feature and lead to an undesirable result.

2) The intention of the first feature is violated, such as the interaction between FW and OCS. This kind of FI occurs when the second feature changes the results of the first feature.

The algorithm runs in FIM. FIM first check if the FI occurs in first case, then check if it occurs in the second case, by executing the following steps:

Step 1. Receive and store FIM-SS messages. If the trigger condition of the second feature arrives before end of the session, go to next step; otherwise exit (there is only one feature running);

Step 2. Modify the trigger condition of the second feature with the initial session parameters and send to FRS;

Step 3. Store FIM-SS and FIM-FRS messages simultaneously until the desirable result of the second feature running in FRS and its actual result running with the first feature in AS are both received;

Step 4. Compare the two results obtained in Step 3. If they are different, show that the intention of the second feature is violated and FI occurred.

Step 5. Draw the trigger conditions of the first feature from initial parameters and send to FRS.

Step 6. Wait for the desirable result of first feature running in FRS. Compare with the actual result getting in Step 3. If they are different, show that the intention of the first feature is violated and FI occurred.

IV. EXPERIMENT

Experiments have been done by emulating the logical entities in Fig.2, such as AS, SS, FRS, UA, or FIM, etc., each with a PC in an IP network environment. The open-source SIP protocol stack ‘oSIP’ is employed, which provides SIP parser, URL parser, SDP parser, finite state machine (FSM) and some tools. Above oSIP protocol...
stack, an open-source network server ‘Partysip’ is used in the logical entities. Features like ONE, NP, UPT (universal personal telecommunication), CF, OCS, TCS (terminating call screening), etc., have been integrated into AS and FRS.

Fig.3 shows the result of detecting the interaction of CF and OCS. Assume a user d1 with SIP URL of 'sip:a@127.0.0.1' activates the feature OCS, forbidding a call to d2 with SIP URL of 'sip:aa@127.0.0.1:5020', and user d1 activates feature CF, forwarding all the call to d2. When d1 calls d2, it forwards the call to d3, so d1 can still make a call to d2.

In this case, the intention of OCS is violated. The FI belongs to second case as is stated above, and is found in the Step 6 of the algorithm.

V. CONCLUSION

In this paper, a method for FI detection is proposed. The method detects FI during the feature work, and needn’t know the feature logic in advance. Thus it is suitable to apply to NGNs. The method focus’ on the intention-violation FI and has been implemented in a simulating NGN networks. Though the method is practicable, it has two disadvantages: one is the increase of cost due to adding of FRS; another is that, if FRS must communicate with SS, the disadvantages: one is the increase of cost due to adding of FRS; another is that, if FRS must communicate with SS, disad.

REFERENCES


513