Service Broker for Managing Feature Interactions in IP Multimedia Subsystem

Anahita Gouya, Noël Crespi
{anahita.gouya, noel.crespi @int-evry.fr}, Institut National des télécommunications (GET-INT)
Mobile Networks and Multimedia Services Department, 9 Rue Charles Fourier, 91011, Evry Cedex, France

Abstract
Since the vast deployment of telecommunication services, until the broad development of innovative 3G services, dealing with Feature Interaction issue has been constantly challenging. This issue results in the occurrence of incorrect or unexpected behaviours from the features and services that are invoked during an IP multimedia session. The purpose of this article is to include a flexible SIP-based Feature Interaction Detection and Resolution algorithm in the service invocation mechanism of IMS. Through several feature interaction examples, we confirm that our proposed algorithm covers a wide range of Feature Interactions and manages the conflicts that arise between various features.

Key words: *Feature Interaction, Interaction Detection and Resolution, IMS, SIP.*

1 Introduction

Today, network operators and service providers offer to their subscribers a wide range of innovative 3G services, such as video calling, Multimedia Messaging, Multiparty Gaming and Mobile TV. Moreover, service providers compete for proposing ubiquitous and real time multimedia services to their users. Consequently, users take advantage of customized and user-centric services. However, despite of the remarkable developments brought by these advanced services, yet, managing interactions between various *features* of these services is a challenging issue.

The term of *feature* has been used in traditional telecommunication services for referring to an independent service unit that network offers to subscribers in order to enrich the offered services. Call Forwarding and Originating Call Screening are examples of features. Call Forwarding allows an incoming call to be redirected to a called-party defined destination. Originating Call Screening screens outgoing calls to destinations that are in black list.

Feature Interaction occurs when each of the features behave correctly separately and independent of each other, but not when running together. For example, suppose that Alice wants to establish a video call with Bob: Alice has an Originating Call Screening feature that blocks outgoing calls to end users who are in black list. On the other hand, Bob has a Call Forwarding Unconditionally feature that diverts all the incoming calls to Anne. But, what happens if Anne is in black list of Alice? In this case, an interaction occurs between Originating Call Screening feature of Alice and Call Forwarding Unconditionally feature of Bob: If Originating Call Screening works correctly as defined for Alice, then Call Forwarding Unconditionally will be neglected. Additionally, if Call Forwarding Unconditionally behaves as it is defined for Bob, then Originating Call Screening feature of Alice will be ignored.

Feature Interaction management issues have been widely studied in the traditional telephony services; multiple research efforts have been performed and different Feature Interaction Detection and Resolution mechanisms have been proposed. However, since the specification of IMS [1] (IP Multimedia Subsystem) by 3GPP as the standard common all-IP service control layer over UMTS, dealing with Feature Interaction issue in IMS still remains challenging.

The goal of this paper is to propose a Feature Interaction Detection and Resolution algorithm that improves this shortcoming in IMS. The remainder of this article is organized as follows: Section 2 overviews Feature Interaction problem and presents the related research works proposed for managing Feature Interactions. In section 3, we introduce service invocation mechanism of IMS specified by 3GPP and we discuss the requirements for providing this mechanism with a Feature Interaction Detection & Resolution algorithm. In Section 4, we present our proposed Feature Interaction Detection & Resolution algorithm to be included in IMS service invocation mechanism. Moreover, several use cases of the proposed algorithm are depicted. In Section 5, we summarize the lesson learned and conclude by referring to perspectives to our work.

2 Overview of Feature Interaction Issue

Feature Interaction refers to incompatibilities and conflicts that happen between features that are invoked during a multimedia session. Feature Interaction may occur between features of one end party (caller *or* callee) or between features of different end parties (caller *and* callee). We denote these cases respectively as *Intra domain Feature Interaction* and *Inter domain Feature Interaction*. For example the incompatibility between Call Barring and Operator Service features causes an Intra domain Feature Interaction. Suppose that Alice invites Bob to join a Multiparty Gaming. Nevertheless, Alice has a Call Barring feature that restricts her outgoing calls to Bob. If Alice uses an Operator Service feature that enables her to call Bob indirectly (by the intermediate of the operator), then the Call Barring feature will not work as it was supposed to.

As an example for Inter domain Feature Interaction, we can refer to the conflict between Caller ID

Hiding feature of Alice and Automatic Call Back on Busy feature of Bob. Caller ID Hiding enables Alice to mask her ID on the outgoing calls. If Alice activates this feature, then the Automatic Call Back on Busy feature of Bob will not work correctly.
Interaction between Originating Call Screening feature of Alice and Call Forwarding Unconditionally feature of Bob, discussed previously is another Inter domain Feature Interaction example. Hence, as presented in these examples, simultaneous invocation of features (that work accurately separately), may lead to erroneous and unexpected behaviors of the features.
Considerable research works have been performed to deal with Feature Interactions management issue: [4 and 5] highlight different reasons for the occurrence of Feature Interactions. [6] proposes a Feature Interaction avoidance mechanism in which the end users negotiate their available services before call establishment. [7 and 8] present the solutions for detecting and resolving the Feature Interactions based on predictable and predefined Feature Interaction detection and resolution mechanisms. These mechanisms are called Offline while they are performed before that features are invoked. Nevertheless due to the unpredictable behaviour of services as well as the vast introduction of new services, not all Feature Interactions can be detected and resolved by Offline methods. [9 and 10] propose mechanisms to detect the Feature Interactions Online i.e. at the service run time. However these propositions do not cover Online Feature Interaction resolution issues.
In the next section after a brief presentation of the service invocation mechanism in IMS we outline the shortcomings of this mechanism for dealing with Feature Interaction management issue. Furthermore, we present the requirements for modifying IMS service invocation mechanism and providing it with both Online and Offline Feature Interaction detection and resolution algorithm.

3 Requirements for Providing IMS with Feature Interaction Detection & Resolution Algorithm

In IMS, Multimedia session establishment, modification, control and termination are performed through SIP [3] (Session Initiation Protocol). Among different SIP proxies included in the functional architecture of IMS, S-CSCF (Serving-Call Session Control Function) is regarded as the brain of IMS for session controlling and service invocation. S-CSCF evaluates the IMS subscriber's service profile to find out which Application Server must be invoked [2]. However, the current IMS service invocation mechanism ignores Feature Interaction issues:

- Intra domain Feature Interactions are not evitable. Whereas IMS subscriber's service profile represents only a static list of service invocation rules. It does not consider different behaviors of services invoked during the session establishment and conflicts between a service to be invoked and the already invoked services will not be managed.
- Inter domain Feature Interactions are not avoidable neither. While the end parties have no idea about the services that have been/will be invoked for the other end party. Hence, the conflicts between these services are not preventable.

To handle these shortcomings, we propose to add a Service Broker over S-CSCF containing our proposed SIP based Feature Interaction Detection & Resolution algorithm. Delegating the Feature Interaction management functionalities to Service Broker (and not services themselves) ensures the autonomy and independency of services and enables their independent development.
Based on our proposed algorithm, all through the session establishment time, Service Broker will gradually be provided with service information that assists to detect and resolve the eventual Feature Interactions. This information comprises the already invoked services during IP multimedia session and the services not to be invoked. The extensibility of SIP enables feasible realization of such modifications and improvements in IMS. Before presenting our Service Broker in detail, we introduce the required Feature Interaction management functionalities of this algorithm:

1. *Offline Feature Interaction Detection & Resolution: Service Identifier Comparison*

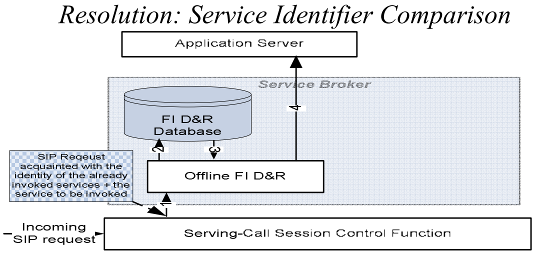

Figure 1: Offline FI D&R Functionality of Service Broker

As illustrated in figure 1, before that each Application Server is invoked, Offline Feature Interaction Detection and Resolution functionality of Service Broker verifies if the Application Server to be invoked is compatible with the previously invoked Application Servers or not. For enabling such verification, the SIP request must be provided with the identity of the already invoked services (during the IP multimedia session) as well as the identity of the service that must be invoked. This verification is based on a predefined Feature Interaction Detection & Resolution database provided over Service Broker, indicating which features are in conflict. Moreover this database defines Feature Interaction resolution mechanism for the detected Feature Interactions.

2. *Online Feature Interaction Detection & Resolution: Service Rule Comparison*

As illustrated in figure 2, after each service invocation, Application Server should provide the Online Feature Interaction Detection and Resolution algorithm of Service Broker with service information that indicates which behaviors (of the next services that will be invoked) are not acceptable. This indication will be presented in form of "Service Rules" and "Service Identifiers".

"Service Rule" information enables the Online Feature Interaction Detection and Resolution functionality of Service Broker to verify if the invocation of the next service is compatible with Service Rules defined by the previously invoked services. In order to prevent the introduction of abusive rules from the Application Server, Service Broker verifies if the Service Rule defined by Application Server is acceptable or not. This verification is based on Unauthorized Rules initially defined by the network over Service Broker.

"Service Identifier" information enables the Offline Feature Interaction Detection and Resolution functionality to verify if the next Application Server to be invoked (based on the service profile evaluation performed by S-CSCF) is not in conflict with the services that have already been invoked and have added their identifier on the SIP message.

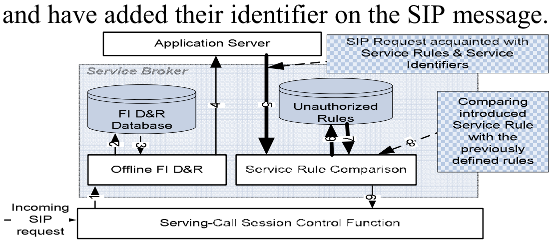

Figure 2: Online FI D&R Functionality of Service Broker, Service Rule Comparison

3. *Online Feature Interaction Detection & Resolution: Favourite Application Server Reinvocation:*

Once all the services of an end party are invoked (based on the service profile), Service Broker must verify that the previously invoked "Favourite Application Servers" are agree with the last SIP message received from the last invoked service. By "Favourite Application Servers" we mean services that are defined by user or network as privileged services that need to be aware of the last SIP message results from the successive service invocation. This functionality of Service Broker is illustrated in figure 3.

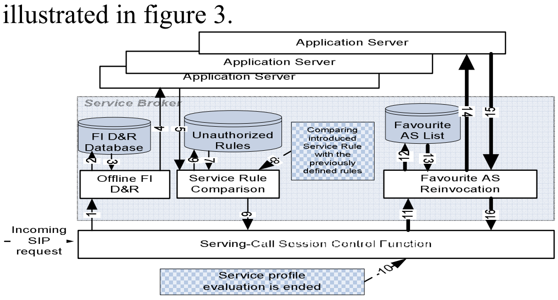

Figure 3: Online FI D&R Functionality of Service Broker, Favourite Application Server Reinvocation

4 Feature Interaction Detection and Resolution Modules over Service Broker

Based on the required functionalities defined in the previous section and as illustrated in figure 4 we provide the Feature Interaction Detection & Resolution algorithm of Service Broker with the following modules:

- Service Identifier Comparison module
- Service Rule Comparison module
- Favourite Application Server Reinvocation Management module

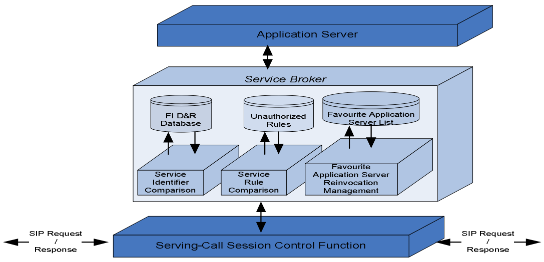

Figure 4: Proposed Modules for Service Broker in IMS

4.1 Service Identifier Comparison module:

This module verifies if the Application Server to be invoked is in conflict with the already invoked Application Server(s) or not. In order to realize this comparison, a unique identifier must be associated to each Application Server. [12] defines a global identifier to be associated to each IMS communication service. Alongside with this definition, in order to realize the Service Identifier Comparison module of Feature Interaction Detection and Resolution algorithm we propose:

- To associate a unique Service Identifier (ID) to each of the widely deployed services.
- To add an extended SIP header called "Serv-ID" to SIP message enabling Application Server to add its ID in Serv-ID header of SIP message.
- To define a Feature Interaction Detection and Resolution Database over Service Broker containing a list of a limited set of Feature Interactions that can be detected and resolved statically and offline. Also, this database will be provided with "Application Server Address – Service ID" association that allows Service Identifier Comparison module to retrieve the identity of the service to be invoked.

Therefore, based on these propositions and once Service Broker invokes Service Identifier Comparison module, this later compares the content of the already added Serv-ID headers (from the previously invoked services in the IP multimedia session) with the ID of the service to be invoked.

If based on Feature Interaction Detection and Resolution Database, the Application Server to be invoked is in conflict with the already invoked services (indicated in the Serv-ID headers of the SIP message); this module resolves the detected Feature Interaction as defined in the Database.

➢ *Use Case: Interaction between Call Forking and Voicemail*:

Suppose that a network operator defines by default Call Forking feature over S-CSCF. In fact, S-CSCF as a SIP Proxy can be defined in a manner that on reception of "No Answer" or "Busy" from callee, forks and sends the incoming SIP request to other available SIP addresses of callee. Hence, if Alice calls Bob on his cell phone and there is no answer from Bob, S-CSCF forks to reach Bob on an available SIP address (For example on his office phone). However, if Bob has Voicemail service that saves the incoming messages once his cell phone does not answer or is busy, by evaluating the service profile of Bob, S-CSCF recognizes that on the reception of SIP "No answer" message, voicemail must be invoked. In this case, Call Forking and Voicemail are in conflict and consecutive invocation of these features will result unexpected and incorrect behaviors. As a result of this interaction, the call of Alice may be diverted to the office phone of Bob and not on his voicemail!

Service Identifier Comparison module as we propose on Service Broker will detect and resolve this Feature Interaction as following:

After Voicemail Application Server invocation, the SIP message that S-CSCF receives from Voicemail will contain a Serv-ID header indicating the identity of Voicemail Application Server. Then, once S-CSCF sends the forking invocation request to Service Broker, this later invokes Service Identifier Comparison module, where based on Feature Interaction Detection and Resolution Database the compatibility between Voicemail (already invoked feature) and Call Forking (feature to be invoked) will be verified. In fact, network provides Service Broker with a Database containing a list of predictable and predefined Feature Interactions, as well as the mechanisms for resolving these interactions. In the database, interaction between Call Forking and Voicemail is predefined and a resolution method is expressed. For instance, a resolution method for the detected Feature Interaction can be: "ignore Call Forking feature". Hence, according to the interaction resolution method, Service Broker ignores Call Forking invocation request and sends back the SIP message to S-CSCF which sends the message to Voicemail Server.

4.2 Service Rule Comparison module:

Not all Feature Interactions can be detected and resolved by the proposed Offline method. The Service Rule Comparison module that we define here introduces Online Feature Interaction detection and resolution mechanism on Service Broker. This module enables Application Servers to include Service Rules to SIP message during service invocations. Consequently, next invoked Application Servers must respect the rules defined by the previously invoked services. We define this module based on the Service-Rule proposition specified in [11]. In this proposition, each invoked Application Server is able to add "Service-Rule" headers to the SIP message indicating which modifications (by the next invoked services) on the SIP message are not accepted for this Application Server. The proposed syntax for the Service-Rule header is as following: *Service-rule: [Applicability]; [messagePart]; [forbiddenValues]*

This SIP header indicates which values (forbiddenValues) are not accepted over which elements (messagePart) of which SIP message (Applicability).

However, defining abusive rules from the Application Servers must be controlled and avoided. Hence, our proposed Service Rule Comparison module verifies if the rule(s) defined by Application Server is (are) compatible with:

1) "Unauthorized rules" i.e. rules initially defined by network over Service Broker for preventing Application Server to define abusive rules

2) Previously defined rules i.e. rules defined by previously invoked Application Servers

Once Service Broker invokes Service Rule Comparison module, following functionalities will be performed:

I. *Comparing Service Rule(s) defined by Application Server with unauthorized rules:*

If the defined Service Rule is not accepted, this module drops the SIP message received from Application Server. (Service Broker continues session establishment procedure by neglecting the SIP message received from Application Server and by considering the SIP message that has been sent to Application Server)

II. *Comparing Service Rule(s) defined by Application Server with the previously defined rules:*

If defined Service Rule is not compatible with the previously defined Service Rules, this module drops the SIP message received from the Application Server.

III. *Comparing the Service Rules defined by caller with the unauthorized rules of the callee:*

Once a SIP message arrives to the callee, Service Broker invokes Service Rule Comparison module for controlling the compatibility of incoming Service rules (from caller) with the local unauthorized rules of the callee. If they are not compatible, this module sends an error message to the caller.

Related to each of the defined functionality, we present a Use Case.

- *Use Case: Feature Interaction due to neglecting the unauthorized rules:*

Suppose that the following unauthorized rule is defined on Service Broker: *Service-Rule: Applicability= 181; messagePart= requestURI, To; ForbiddenValues = all.*

This rule indicates that forwarding the incoming call is not allowed (SIP 181 message refers to: "Call is being forwarded"). Hence, if Application Server intends to forward incoming call and sends a "SIP 181 message" to Service Broker, Service Rule Comparison module of Service Broker will reject this message and refuse the forwarding.

- *Use Case: Interaction between Originating Call screening (OCS) and Call Forwarding Unconditionally (CFU):*

If Alice wants to Call Bob, when Bob has a CFU feature for forwarding all the incoming calls to Anne, and that all calls from Alice to Anne are screened, an interaction occurs between OCS feature of Alice and CFU feature of Bob. Nevertheless, according to the Service Rule Comparison module that we introduced over Service Broker, when OCS feature is invoked, the outgoing SIP message sent from OCS to S-CSCF will contain following header: *Service-Rule: Applicability= INVITE; messagePart=TO; ForbiddenValues =Anne.*

This header indicates that all SIP INVITE requests to Anne are forbidden. Then, when S-CSCF of Bob invokes CFU feature, in order to divert the incoming call to Anne, Service Rule Comparison module over Service Broker of Bob, detects that the SIP request received from CFU (i.e. INVITE Anne) is in conflict with the service rule defined by OCS (no INVITE to Anne is allowed). Hence, Service Rule Comparison module rejects the "INVITE Anne" request and sends an error message to Alice.

- *Use Case: Interaction between Caller ID Hiding and Automatic Call Back on Busy:*

Suppose that Alice wants to call Bob. Alice has a Caller ID Hiding feature by which she can hide all outgoing calls. On the other side, when a call arrives to Bob and this later is busy, as Bob has an Automatic Call Back on Busy feature, S-CSCF of Bob must call back Alice once Bob is no more busy. But since Alice has hidden her ID, Call Back on Busy feature will not work correctly.

The Service Rule Comparison module that we proposed detects and resolves this Feature Interaction as following: Service Rule Comparison module of Service Broker over S-CSCF of Bob is provided with the following unauthorized rule: *Service-Rule: Applicability = INVITE; messagePart= From; ForbiddenValues = Anonymous.*

This rule indicates that calls from anonymous are rejected. Hence, once the SIP message arrives to Service Broker of Bob, the Service Rule Comparison module detects that the INVITE request (INVITE from anonymous) is in conflict with the local unauthorized rule of Bob (No INVITE from anonymous is allowed). Therefore, this module rejects the INVITE request and sends an error message to Alice.

4.3 Favourite Application Server Reinvocation Management module:

Successive Application Server invocations by S-CSCF may result in unexpected modifications over SIP message and consequently Feature interactions may occur that have not been detected and resolved by the previously defined modules. Our proposed Favourite Application Server Reinvocation Management module deals with this issue as following:

Network or the user can provide Service Broker with a list of Favourite Application Server(s) that need(s) to be aware (and agree) with the last SIP message results from the successive service invocations. Once all services of an end party are invoked (based on the service profile), Service Broker invokes the Favourite Application Server Reinvocation Management module in order to compare the last SIP message received from the last invoked Application Server to the SIP messages received from the "Favorite Application Server".

In order to compare the SIP messages, we use "Dialogue State" that as specified in [3] is composed of: dialog ID, Local Sequence number, Remote Sequence number, Local URI (from/to), Remote URI (from/to), Remote Target (Contact headers), Boolean "secure" (SIPS, TLS) and Route set of SIP message. In fact, the parameters defined by Dialog State are adequate for enabling to recognize the modifications performed over the SIP messages and consequently the reason for choosing Dialog State for comparing SIP messages is that it contains satisfactory information about the characteristics of the current IP multimedia session.

By each service invocation, the Favourite Application Server Reinvocation Management module verifies if the Application Server to be invoked is a Favourite Application Server or not. If yes, then this module saves the Dialog State of the SIP message received from this favourite Application Server. Afterwards, by the end of service invocations, this module compares the last message received from the last invoked Application Server with the message received from favorite Application Server(s). If they are not identical (different Dialog State), then this module reinvokes the Favourite Application Server(s) in order to verify if they are agreeing with this final SIP message. Favourite Application Server must be able to distinguish between the first invocation and the reinvocation and further decisions (in case of the interaction occurrence) are performed by favorite Application Server.

- *Use Case: Interaction between Call Barring and Prepaid and Operator Service Features*

Suppose Alice wants to perform an international call to join Bob by using a Prepaid service. But Call Barring feature of Alice restricts outgoing calls to international destination. However, Alice uses an operator assisted service to join Bob. Therefore even if Call Barring feature had controlled that Alice is not calling an international number the Operator Service feature neglects these controls and performs an international call for Alice.

Favourite Application Server Reinvocation Management module that we introduce over Service Broker detects and resolves this Feature

Interaction: Network defines Call Barring and Prepaid services as Favourite Application Servers. Then, Favourite Application Server Reinvocation module over Service Broker will compare the Dialog State of SIP messages received from these Application Servers with Dialog State of the SIP message received from Operator Service and recognizes that the Dialog States are not identical (different SIP URIs: the last SIP message is destined to an international number, while the first two messages contain a local number). Therefore Favourite Application Servers will be reinvoked and consequently Call Barring feature will not comply with final SIP message (while international calls are forbidden). Hence the message will be rejected.

4.4 Feature Interaction Detection and Resolution Algorithm:

Based on the defined modules, we propose to include following algorithm to the service invocation mechanism of IMS:

(Modifications to the current service invocation mechanism are stated in *italic*).

Step1: S-CSCF receives an initial SIP request, starts to evaluate service profile of user and recognizes that an Application Server must be invoked.

Step2: *Before invoking the Application Server, S-CSCF sends the invocation request to Service Broker function*.

Step3: *Service Broker invokes the **Service Identifier Comparison** module*: To verify the compatibility of the Application Server to be invoked with the already invoked Application Servers; based on the Feature Interaction Detection and Resolution Database.

*On callee side, Service Broker will then invoke the **Service Rule Comparison** module*: To control the compatibility of incoming Service Rules (from caller) with local unauthorized rules defined in callee domain.

Step4: S-CSCF invokes the Application Server. This later sends back a SIP message *containing its identifier in Serv-ID header and the eventual Service Rules in Service-Rule header.*

Step5: *Service Broker invokes the **Service Rule Comparison** module*: To verify the compatibility of Service Rules added by Application Server, with unauthorized rules and with the Service Rules defined by the previously invoked services.

Step6: S-CSCF resumes the procedures from step 2, until the end of service profile evaluation, then:

Step7*: Service Broker invokes **Favourite AS Reinvocation Management** module:* To control if the Favourite Application Server(s) is (are) agree with the last SIP message received from the last invoked service

Step8: Service Broker sends the SIP message back to S-CSCF.

5 Conclusion and Perspectives

In this article we focused on Feature Interaction management issue as a major shortcoming in the current service invocation mechanism of IMS.

In order to deal with this issue, we proposed a SIP-based Feature Interaction Detection and Resolution algorithm. We included this algorithm in the service invocation mechanism of IMS in order to cover Offline and Online Feature Interaction management methods: a variety of Feature Interactions will be managed Offline (before service invocation) and the rest will be detected and resolved Online (during the service invocation).

We justified the convenience of our algorithm by means of various Feature Interaction examples.

In the next step of our research work, we will evaluate the impact of including the proposed Feature Interaction management algorithm into the service invocation mechanism of IMS. This evaluation consists particularly on assessing session establishment delay while the Feature Interaction Detection and Resolution algorithm is included in the service invocation mechanism of IMS.

Other perspective to our work is to extend this algorithm in order to consider inter-operator agreements for enabling privacy control during exchange of service information (i.e. Service Rules defined by the invoked services and the identifier of the invoked services) between end parties.

References

[1] 3GPP, "IP Multimedia Subsystem (IMS)", TS 23.228

[2] 3GPP, "IP Multimedia session handling; IM call model", TS 23.218

[3] IETF RFC 3261, "SIP: Session Initiation Protocol"

[4] M. Calder et al, "Feature Interaction: A Critical Review and Considered Forecast", Elsevier, May 2002.

[5] D.O.keck et al, "The Feature and Service Interaction Problem in Telecommunications Systems: A Survey", IEEE Transactions on Software Engineering, October 1998.

[6] M. Kolberg and E.H.Magill, "Handling Incompatibilities between Services deployed on IP-based Networks", IEEE Intelligent Networks 2001

[7] A. Khoumsi, "Detection and Resolution of Interactions between Services of Telephone Networks", interne report, University of Montreal, 1997

[8] J.Pang, L. Blair, "An Adaptive Run Time Manager for the Dynamic Integration and Interaction Resolution of Features", IEEE ICDCSW 2002

[9] Z. Chentouf et al, "Implementing online Feature Interaction detection in SIP environment: early results", IEEE ICT 2003

[10] Z. Chentouf et al, "Mapping SIP onto a Feature Interaction Management Language", ConTEL 2003

[11] Crespi N. "A distributed mechanism to resolve dynamically Feature Interaction in the UMTS IP Multimedia Subsystem", 6th International Workshop on Applications and Services in Wireless Networks, 2006.

[12] 3GPP, "Identification of Communication Services in IMS", TR 23.816