A distributed mechanism to resolve dynamically Feature Interaction in the UMTS IP Multimedia Subsystem

Noël Crespi
GET-INT, Institut National des Télécommunications. 9,rue Charles Fourier 91011 Evry Cedex FRANCE
noel.crespi@int-evry.fr

Abstract: Feature interaction (FI) or service interaction term is used to indicate conflict between services once they are used together. The FI issue has been widely studied for the Intelligent Network (IN). The IP Multimedia Subsystem (IMS) is the main evolution of UMTS Core Network, it allows to offer converged multimedia services. The service architecture in the IMS is totally different from the one in the legacy networks and solutions proposed for solving FI issue in the IN cannot be applied to the IMS case. The lack of existing FI mechanism in the IMS forces manufacturers to build services in the same functional entity or use proprietary mechanisms. This is a show-stopper for network operators who want to have the possibility to deploy each service independently. In this paper we propose a new distributed mechanism resolving dynamically service interaction in the IMS, even in a multi-domain context.
Key words: SIP, IP Multimedia Subsystem, Service Interaction, SCIM.

1 Introduction

3GPP (Third Generation Partnership Project) standardisation body [1] specifies the IP Multimedia Subsystem in the Release 5 and 6 of the UMTS system to provide IP based multimedia services. Like in the Intelligent Network (IN), service control in IMS is separated from the basic call control and services can be provided by third-parties. SIP (Session Initiation Protocol) [2] is adopted as the signalling protocol to control multimedia services in the IMS.

Having an open architecture allowing third-parties to provide services and access to the network resources facilitates the creation and the deployment of new services. However, the feature interaction issue is likely to become more complex in this environment. The problem of FI has been largely studied in the IN context and many articles has been written and till now no ultimate solution exist. In absence of correct FI mechanism, IMS prototypes (no real IMS exists today) provide all service in the same functional entity, a SIP Proxy or an Application Server (AS) or use proprietary mechanisms. Network operators however expressed the need for a distributed architecture, based on open interfaces, with distinct entities for different services. To deploy new services, network operators should not be tied to a specific manufacturer.

Moreover, the standardization bodies in charge of IMS services, mainly the 3GPP and the OMA (Open Mobile Alliance), do not specify services, but rather functional service building blocks, or Service Capabilities. These Service Capabilities are reusable at runtime by various services, in order to enable building innovative and evolving services mostly independently of network considerations [15]. Examples of Service Capabilities are Presence, Messaging, and multimedia conferencing.

The requirement to have a distributed architecture imposes to solve interaction between services and also between Service Components. The solution we propose in this paper is an answer to this essential question and a key enabler to deploy IMS.

When executing more than one service in the same call or session, a feature interaction occurs when one or more of those services do not behave as expected, i.e. when it is executed separately. The methods to resolve the Feature Interaction issue in telecommunication networks can be divided into tow types:

- *Offline methods*: consist on detecting and resolving the FI before deploying the services in the network. Most approaches to solve the FI in telecommunication networks belong to this category and many efforts have been done in this field. Hence many mature solutions exist even for the SIP-based services. [12].
- *Online methods :* offline methods cannot resolve all of the FI. This is why an online resolution is needed to solve FI dynamically. These methods are tightly related to the underlying service control architecture.

In SIP-based networks the feature interaction issue may be more complex than in a traditional network and that was recognized by H. Shulzrinne [4], one of the prime authors of SIP. However, as SIP is an expressive and extensible signalling protocol, the resolution of feature interaction may be easier. The work on the feature interaction in SIP-based networks has started and several approaches have been proposed [5,6,7]. However those solutions are not adapted to the IMS case.

In [5, 6] the proposed solution to detect and resolve the feature interactions between services is based on the definition of a language to describe SIP services and on a centralized entity which possesses the description of the deployed services. However this solution is not adapted to the IMS case because interacting services may be distributed in two domains: the caller and the callee domains. We shall also note that operators do not wish to reveal to third-parties or to other network operators the detail of the service they provide to their subscribers.

In [7], an approach based on negotiation between the caller and the callee to avoid service interaction is proposed. The INVITE message is extended by a list of possible actions the callee may perform if it is not capable of accepting the invitation directly. If the callee can accept the INVITE immediately, no further actions are required and the extended INVITE request is treated as a normal INVITE message. However, if the callee cannot accept the call directly, it should look through the list of acceptable options. If the callee finds one of the options acceptable, it should perform that action. However, if no action is acceptable, the callee will decline the invitation or propose other actions via a new proposed SIP message called SUGGEST. This solution is destined to avoid the feature interaction only between services residing in the user side and does not take in account the services provided by the network that reside in Application Servers. Moreover the need for an additional SIP method (SUGGEST) is a considerable drawback for this mechanism.

In this paper we propose a mechanism to avoid feature interactions between services in the IMS. We have chosen the IMS case because it the most complete standardised architecture existing today for SIP real-time multimedia services. It is also applicable from an xDSL access as defined in the TISPAN architecture [11] which is based on IMS. Our mechanism uses the expressivity and the extensibility of the SIP protocol to avoid the feature interactions. Unlike most of proposed solutions, our mechanism is dynamic and consists of online FI avoidance.

The rest of this paper is structured as follows: In section 2 we present IMS service architecture. In section 3 we analyse the service interaction issue in the IMS case. In section 4 we describe our proposed mechanism to avoid the FI. And finally we conclude in section 5.

2 Service Architecture in the IMS

3GPP specified IMS as an open and standard IP-based infrastructure over UMTS core network for providing multimedia services in NGN. The IMS is open to third party services; in order to facilitate the service provision by third-parties the 3GPP has joined the PARLAY group [8] to define an open Application Programming Interface (API), called OSA [9]. OSA provides to third-parties a secure access to the network capabilities. The OSA interface allows service developers to use the capabilities offered by the network in order to create services without manipulating the underlying protocols.

CAMEL services can be accessed via the IMS by using the IM-SSF (IP Multimedia Service Switching Function) [10]. The role of the IM-SSF is to allow reusing legacy IN services such as prepaid.

Consequently the service logic in the IMS can be hosted by three types of Application Servers:

- SIP AS
- OSA AS
- CAMEL Server via IM-SSF.

OSA and CAMEL environments do not have a native SIP interface. The OSA Service Capability Server (SCS) performs mediation between the S-CSCF (Serving Call Server Control Function) and OSA-based ASs. The IP Multimedia Service Switching Function (IM-SSF) performs mediation between the S-CSCF and CAMEL Server: it provides a mapping between SIP/ISC and CAMEL Application Part (CAP) protocols. The ISC interface between the S-CSCF and IM-SSF, OSA SCS and SIP-based application servers is based on SIP. From the S-CSCF, all Application Servers are therefore based on SIP.

The mechanism proposed in this paper to solve FI is applicable to the three types of Application Servers: SIP AS, OSA and CAMEL.

2.1 Services triggering mechanism

During the registration phase, an S-CSCF is assigned to control the user's services; the user profile is downloaded from the HSS to the S-CSCF. It contains Initial Filtering Criteria (iFCs), structured in an XML format that allows the S-CSCF to decide which service(s) shall be invoked during a SIP session or transaction and in which order they should apply [1].

Each iFC contains the conditions to be satisfied in order to trigger a specific service, the name of the Application Server hosting the service, optional service related information to be sent during the invocation and the priority of the service among the other services. When the S-CSCF receives a SIP request matching the iFC, it invokes the associated service by forwarding this SIP request to the Application Server indicated in the iFC. iFCs are only applied to initials SIP request i.e. the requests initiating a SIP session or transaction (INVITE, SUBSCRIBE, REGISTER, OPTION…), consequently the service invocation can be done only statically in the SIP session or transaction initiation phase.

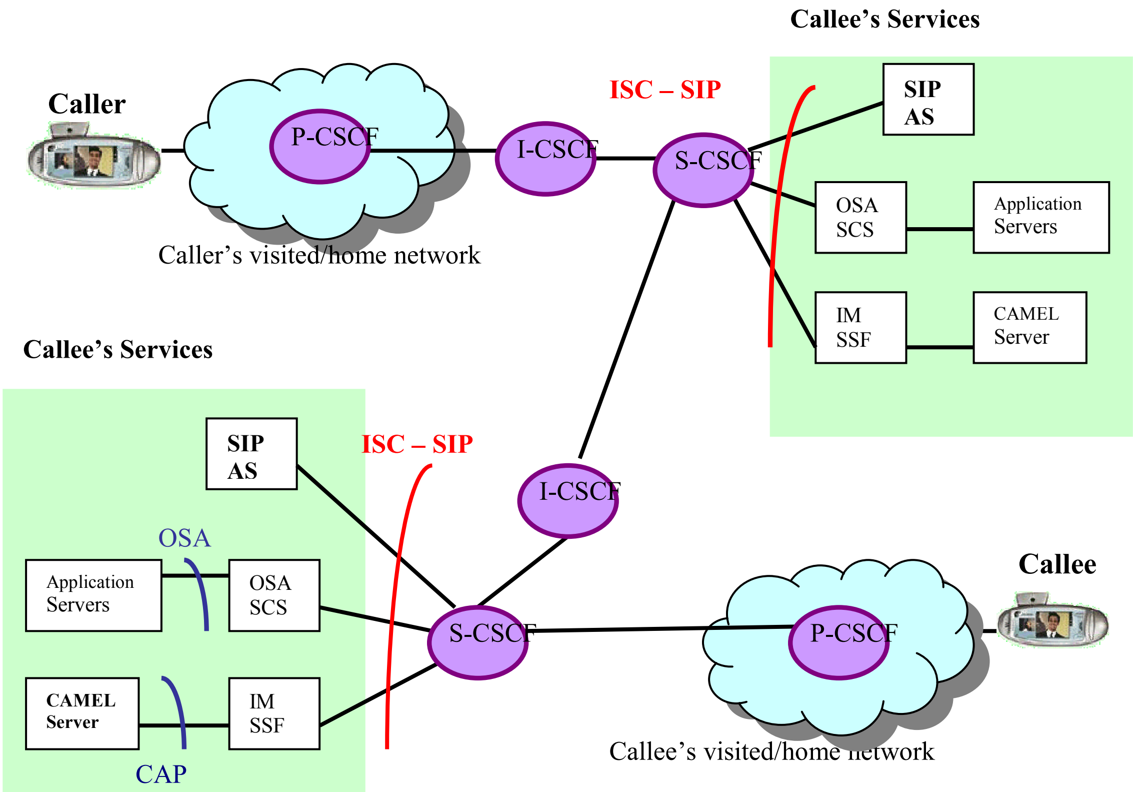

Figure 1: service architecture in the IMS

A user may subscribe to several services, and as a consequence several iFCs may be present in his user profile. When the S-CSCF receives an initial SIP request, it checks whether it matches the iFC that has the highest priority for this user. If it does not match, the S-CSCF checks the next iFC, in the predefined priority order. If it matches, the S-CSCF forwards the request to the indicated AS. This AS executes the service logic, eventually modifies the request, and sends it back to the S-CSCF. The S-CSCF performs the same processing with the next unexecuted iFC. The S-CSCF continues this process until all the iFCs are checked or if an AS ends locally the request as a part of the service logic, e.g. a prepaid account without remaining credit.

This AS invocation mechanism defined in the IMS is therefore fully static and no action can be taken to modify the triggering or to inhibit certain services.

In current 3GPP IMS specifications (Release 5 and Release 6), service control is always performed in the home network. The services of a user are therefore always controlled by a S-CSCF located in the home network even when roaming in a visited network. However, the 3GPP standard states that there will be standardised means to access local services in the visited network [1] [17], but the mechanisms are still unspecified.

3 Service interaction issue in IMS

In a SIP session involving two end users, there may be two S-CSCF involved in the SIP session or transaction: a S-CSCF controlling originating services for the caller and another one controlling terminating services for the callee (figure 1). Each S-CSCF may invoke one ore more services in the order predefined in the iFCs. These two S-CSCFs may be located in different networks. From that statement, we can divide IMS service interactions into two groups: interactions between services controlled by the same S-CSCF and interactions between services controlled by different S-CSCFs, which is the most general case.

An S-CSCF should not manage interactions between services: the 3GPP introduced for this purpose a new functional entity, the Service Capability Interaction Manager (SCIM). The SCIM interfaces between the session control layer and the service execution layer and performs the service interaction detection and resolution before the services are invoked. Apart from specific studies [16], the SCIM remains completely unspecified in UMTS Release 6, and service interaction issue is still not resolved.

SIP gives to the end user the possibility to create and manage services. These services reside in the user device. Hence, in the IMS, we can make a distinction between two types of services: user provided services and network provided services. For the first type of services we believe that the approach described and proposed in [7] and the caller preferences information sent in the SIP message which are defined by the IETF in [3], are complementary and can solve all the user provided services interactions. However, these approaches are not adapted to solve interactions involving network-provided services because they focus only on the end user. Our solution is destined to solve the interactions involving network-provided services FI, but can also be used to solve user-provided services interactions.

Having analysed the studies to solve service interactions, especially in the Intelligent Network, we have found that in the offline phase a large amount of work has been done and many formal and informal solutions have been proposed to detect FI. After having detected conflicting interactions between services in this phase, certain FI can be solved by redesigning services or just defining adequate priorities of each service (in the IMS case only). But certain interactions cannot be solved offline and an online resolution is mandatory for a correct deployment for the IMS with distributed services. In the IMS case, we have divided the FIs, which cannot be solved offline into two types:

- **Single component interactions:**

Interactions between services controlled by the same S-CSCF, known in the literature by *Single-component interactions*, which happen only in particular circumstances which depend on the user profile(s) of the involved user(s) in the session. Services are conflicting only in some circumstances and are compatible in the others. For example the interaction between Operator Services (OS) and the Originating Call Screening (OCS) service is single component. An OS service allows subscribers to establish operator assisted calls. The OCS service allows to an OCS-subscriber to screen outgoing calls based on the destination. The screened destinations are defined by the subscriber. There is an interaction if an OCS subscriber having as screened destination X tries to call X using the OS service.

- **Multi-component interactions:**

Interactions between services controlled by different S-CSCF belonging to the same or different domains. This type of interactions is known in the literature by *multi-component interactions* and it is the most complex to solve. Indeed, in this case conflicting services belong to different administrative authorities. The problem comes from the fact that information about the running services in one component lacks to the other component. Therefore, an offline resolution is impossible and the online one cannot be possible if there is no communications about the running services between the two domains. In order to illustrate that, let's take the example of the interaction between the Originating Call Screening (OCS), presented above, and the Call Forwarding (CF) service. The CF service allows a subscriber to forward incoming calls to a specific destination unconditionally or depending on certain conditions (busy, no reply, not reachable). There is an interaction if an OCS-subscriber (A), who screens his outgoing calls destined to C, calls a subscriber (B) who forwards his incoming calls to C. Indeed, in this case the OCS service of A will have not screened the call to C even if this later is in his screening list. This FI can be solved with the FI mechanisms proposed in this paper.

After having analysed most of the known FIs, we have realized that most of the single-components interactions can be avoided simply by choosing a suitable services execution order which is possible by using the priority mechanism existing in the services triggering system of the IMS (see section 2) and the service control mechanism. The two following examples illustrate this statement.

Terminating Call Screening (TCS) and Call Forward on Busy subscriber (CFB): The TCS service allows to the subscriber to screened incoming calls based on the originating address (black list or white list) and other parameter like time and day. A subscribed to TCS and CFB services. For his TCS service, A has specified that B is member of a black list, i.e. calls from B shall be rejected. If B calls A and this later is busy, a conflicting interaction may happen in an IN based network between the two services: the incoming call may be forwarded instead of being rejected. In the IMS case, we can address this issue in giving a higher priority to the TCS service and the interaction will be avoided.

Terminating Call Screening (TCS) and Last Number Redial (LND): LNR service records automatically the last incoming call of an LND-subscriber (A) when the later is busy or has missed an incoming call. A may decide to call the last missed incoming call, the LND service will then establish a call with A and the last caller. If the last caller was a screened user for the TCS service there is a conflicting interaction because the TCS will not have rejected an incoming call from a screened caller. In the IMS, in order to avoid this interaction, it is enough to give the TCS service a higher priority than the LNR service. Thus, when a screened incoming call is received, the TCS service will reject it and the call will not be in the LNR list.

4 Our proposed approach to avoid FI in the IMS

After having analysed known FIs which can not be resolved offline, we have realised that for each service which may have a conflicting interaction with other services we can define some conditions that if they are satisfied there will be no conflicting interactions. These conditions can be defined in the service development or in the offline detection and resolution phase. For instance, in order that OCS service for a given subscriber works correctly, during the session, any other service must not forward the call to a destination existing in the screened list defined by the subscriber. Another example is the Redial service: the condition to be satisfied for this service is: the call should not be forwarded if the callee is busy.

We therefore propose to extend SIP to transport FI information and to specify the associated mechanisms to avoid interactions between services wether they are in the some domain/entity or not.

Service logic applications provide services by impacting and modifying the SIP signalling. The conditions that must be satisfied for a service in order to avoid conflicts with other services can then be translated by describing forbidden modifications of the SIP messages. As exposed in section 1, 3G defines an architecture capable to offer various types of multimedia service without standardizing the service itself. Then, in order to be general and to allow the description of any kind of conditions for current and future services we have adopted a low level description. This description is based on one or more rules. A rule may specify forbidden modifications of one or more SIP message element or specify forbidden value(s) for one ore more SIP message element.

These rules can be added by a service to any SIP request belonging to the related SIP session or transaction. A service may need to create rules that shall be applied only to the sent Request, to one or more of its responses or to the whole transaction. To achieve this, a service logic which adds a rule shall specify the applicability of this rule. Hence, the rules added to a request during its way to the destination must be copied in its responses.

In order to transport these rules we propose to extend the SIP protocol with a new header called: *service-rule.*

4.1 Rules syntax

As we have said in the previous section the syntax should allow describing any kind of rules for a service, to specify to which part of the SIP message it applies and which SIP messages are concerned. We propose to structure a rule as follows: [applicability] ; [messagePart]; [forbiddenValues].

Applicability: specifies if the rule is applicable to the request and/or to one or more responses. It can contain one or more of the following values:
Request: Indicates that the rule shall only apply to the current request.
Transaction: means that the rule shall apply to the whole transaction: request and its responses.
List of responses code: contains a list of response codes indicating the responses where the rule is applicable e.g.: 480, 600.

messagePart: describes the elements of the SIP message that are affected by the rule. A SIP message element can be the Request-uri, a header or the content of the SIP message.

***forbiddenValues* :** specifies the values that the elements indicated in the messagePart party must not be set on. The value *All* means that these elements must not be changed. Perl-like regular expression can be used to specify forbidden values e.g.: *@domain.com

If a rule applies to a response, it means that it is forbidden to drop the response or create another request which does not satisfy this rule, for this session or transaction. In order to illustrate that, let's take the example of a service logic application that wants to add a rule to an INVITE message, indicating that if the destination is busy the session should not be forwarded to another destination. In this case, the rule added to the INVITE message should indicate that it is applicable to the responses 600 (busy everywhere) and 480 (busy here). The syntax of this rule is:
Service-rule: Applicability= 480, 600; messagePart = requestURI, To; ForbiddenValues = all.
In this case the rule indicates that a service logic, when receiving a response containing this rule shall not drop the response and create a new request belonging to the related SIP session or transaction and which does not satisfy the rule present in the response.

It is important to note that the S-CSCF is not affected by the restrictions added to a SIP message: only Application Servers shall apply the rules. Hence, if an S-CSCF *receives* a user-terminating request, which contains a rule restricting modifications in the requestURI, it will ignore this restriction and performs a normal treatment of this SIP request.

4.2 Rules control

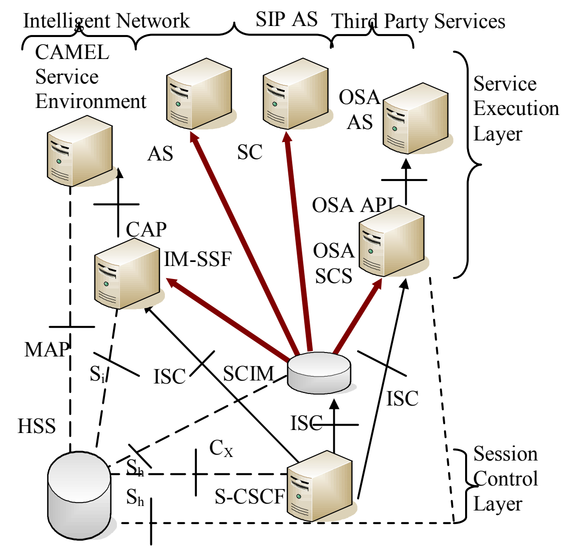

Figure 2: Functional Architecture of IMS

This subsection describes the mechanisms that should be applied to enforce FI control.

A. Rules control in the Network

The 3GPP states the SCIM entity is destined to manage the service interaction only between services hosted in SIP-based application servers. However feature interactions may involve also services hosted in OSA-based AS and CAMEL services. In our solution, we therefore propose to use SCIM to control interactions between all kinds of services.

Each service can specify any kind of rules it wants and the other services involved in the session or the transaction must not transgress these rules. Because the IMS is open to services provided by third-parties, network operators need to control the rules added by the service i.e. check if the service has the right to add a specific rule. It is also essential to control the application of the rules by all any services, in particular when hosted by third-parties: this is one of the roles of the SCIM which is located between the S-CSCF and service platforms. All the messages, sent by the S-CSCF to an Application Server or received from an Application Server, will pass through the SCIM. In the service-provisioning phase of a service, we may provide the SCIM with authorized or unauthorized rules. When a SIP message containing service rules is forwarded by the S-CSCF toward an AS, the SCIM stores the rules before forwarding the request to the AS. When the AS sends back the response, the SCIM checks if the rules have been applied; this ensures that the network operator always controls the service interaction, even when the services are hosted by third-parties. If the rules are not respected, the SCIM drops the SIP response received from the AS and sends to the S-CSCF the copy of the original message. If they are respected, the SCIM checks the rules the AS may have added and suppresses unauthorized rules if any.

B. Rules control for end users

We also need to control that end users do not omit in SIP responses rules originally present in the request. This check is simple and can be done in in the SCIM. When SCIM receives an incoming request it saves the list of Service-Rule headers present in the request before forwarding it to the user. When it receives a response it checks weather the Service-Rule headers has been changed or modified and eventually inserts in the response the Service-Rule headers saved from the original request.

4.3 OSA and CAMEL based Services management:

An OSA service uses the capabilities offered by the network via the OSA API without manipulating the underlying protocols. The SIP extension we propose in this paper can therefore be considered as a new capability offered by the network. Then, in order to allow FI involving OSA based services we have to take in account in the OSA API this new capability. This new capability allows to a service to specify rules that should be satisfied in order to be sure that the features it provides will not be broken by another service. The OSA API should permit to the service developer to describe these rules in an abstract manner. And the OSA SCS gateway will have the role of translating these abstract rules to the syntax we have described in section 4.1. Note that the mapping proposed in this subsection is optional. Next section provides an alternative.

Another approach that can be used for either OSA or CAMEL based services and even for SIP-based services if we want to mask the FI avoidance for the services. This is in particular true for CAMEL as no equivalent of Service-rule exists in CAP, and it is not reasonable to expect it to be standardised, as no significant evolution of CAMEL is expected in UMTS Release 7. This approach consists of using the SCIM entity to insert rules in the SIP requests on behalf of services. Hence, in the service provisioning phase for any kind of service (OSA, CAMEL or in a SIP AS) we can inform the SCIM about the rules that must be applied and to which requests they apply. The SCIM can identify the service via its hosting AS address (a “service key” [10] parameter would be more accurate, but in Release 6, the SCIM does not have this information). Then, when an incoming SIP message generated by a service is received, the SCIM may

insert the rules that shall apply to the service and forward the message to the S-CSCF.
Note that the mechanism proposed in this subsection is optional, and independent of the solution proposed for SIP AS.

4.4 Example of FI avoidance:

Originating Call Screening (OCS) and Call Forward (CF): The OCS service allows its subscriber to define a list of forbidden destinations (FD_list) for originating calls. If an OCS subscriber calls a CF subscriber who has specified to forward his incoming calls to a destination present in the FD_list of the former, there will be a conflicting interaction between the two services. The condition that should be satisfied in order that the OCS service is correctly executed is the call must not be forwarded by any other service toward one of the destination present in the FD_list. Hence, in order to avoid a conflicting interaction with the OCS service, the service logic should add a service-rule header containing a rule indicating that the call must not be forwarded to destinations present in the FD_list. If we assume that the FD_list contains a, b and c, this rule is as follows:
Service-rule: applicability= transaction;
messagePart = RequestURI, To; forbiddenValues = a, b, c
This rule prevents CF service active for the callee to forward the call to destinations a, b or c.

5 Conclusion

Due to its openness and to the use of the SIP protocol, the IMS is expected to allow the deployment of many innovative services. In this article, we address the service interaction issue in the IMS and propose a **distributed mechanism avoiding dynamically all kind of service interactions**: multi-component and single component interaction. The mechanism can apply to originating and terminating services. It can be applied to solve feature interaction event when the two services are in different administrative domains, and does not reveal to other operators the detail of the service. It applies to SIP service in general, and more specifically in the IMS case (for 3G System) and in the TISPAN case (for a xDSL access to the IMS).

We have used the expressivity and the extensibility of the SIP protocol adapted to the service architecture of the IMS: we proposed an extension of the SIP protocol, and the related mechanisms in 3GPP functional entities in order to give to the operator the control of service interaction. This solution is simple to implement and unlike the most of the online proposed solutions in the IN, it does not need large computation in the network, which may have an impact on the session establishment delay.

Based on the mechanism we introduced, we have proposed to use the Service Capability Interaction Manager to control the interaction on the network side, to ensure all Application Server respect the service interaction rule we defined. This is in particular useful with third party Application Servers.

Finally, our proposed solution is capable to avoid interactions of a number of services, and it is not designed to solve only known interactions between existing services, but also interactions that will involve future services.

References

[1] 3GPP TS 23.228: IP Multimedia Subsystem; Stage 2.
[2] IETF RFC3261 "SIP: Session Initiation Protocol", June 2002.
[3] IETF RFC3840, "Indicating User Agent Capabilities in the Session Initiation Protocol (SIP)", August 2004.
[4] Jonathan Lennox, Henning Schulzrinne, Feature Interaction in Internet Telephony, Feature Interactions in Telecom and Software Systems, IOS Press, Amsterdam,2000.
[5] Z. Chentouf, S. Cherkaoui and A. Khoumsi "Mapping sip onto a feature interaction management language", Telecommunications, 2003. ConTEL 2003.

[6] Z. Chentouf, S. Cherkaoui and A. Khoumsi "Implementing online Feature Interaction detection in SIP environment: early results. ICT 2003"

[7] Evan H. Magill "Handling Incompatibilities between Services
deployed on IP-based Networks", IEEE Intelligent Networks 2001, IEEE press, Boston, USA.
[8] http://www.parlay.org/

[9] 3GPP TS 29.198 Open Service Architecture (OSA) Application Programming Interface (API).

[10] 3GPP TS 23.278: Customized Applications for Mobile network Enhanced Logic (CAMEL), IP Multimedia System (IMS) interworking; Stage 2

[11] European Telecom Standards Institute (ETSI), Telephony and Internet converged Services and Protocols for Advanced Networking (TISPAN) group, http://portal.etsi.org/tispan.

[12] Kenneth J. Turner. Modelling SIP Services.
editors, Proc. Formal Techniques for Networked and Distributed Systems (FORTE XV), pages 162-177, LNCS 2529, Copyright Springer-Verlag, Berlin, November 2002.

[13] Gouya A., Crespi N. "Inter-Operator Cooperation Challenges in SIP-Based Service Architecture of IP Multimedia Subsystem of UMTS", Sixth IEE International Conference on 3G & Beyond, 3G 2005.
[14] E. J. Cameron, N. D. Griffeth, Y.-J. Lin, M. E. Nilson, W. K. Schure, and H. Velthuijsen, "A feature interaction benchmark for IN and beyond," Feature

Interactions in Telecommunications Systems, IOS Press, pp. 1–23, 1994.
[15] E. Bertin, E. Bury, P. Lesieur, "Intelligence distribution in next-generation networks, an architectural framework for multimedia services", ICC 2004, Paris, June 2004
[16] Gouya A., Crespi N. and Bertin E. "SCIM (Service Capability Interaction Manager) Implementation Issues in IMS Service Architecture", submitted to IEEE International Conference on Communications, ICC 2006.