14

Service Orchestration in IMS

Anahita Gouya and Noël Crespi

CONTENTS

Introduction

During the last decade, the International Telecommunication Union—Telecommunication Standardization Sector (ITU-T) devoted special interest to identifying the requirements of the next-generation network (NGN). Along

with these efforts, the 3rd Generation Partnership Project (3GPP) defined an all-IP service control overlay, called IP multimedia subsystem (IMS) [1], as an instantiation of NGN. 3GPP adopted the session initiation protocol (SIP) [2] as the IMS signaling protocol to handle the end-to-end session establishment between users. IMS is generic architecture for offering multimedia services and enables network and service convergence into a *single* service platform. In 2004, the European Telecommunications Standard Institute Telecom and Internet Converged Services and Protocols for Advanced Networks (ETSI TISPAN) group initiated providing further inputs to the NGN model of ITU-T in order to migrate the legacy circuit-switched networks to NGN by leveraging the 3GPP IMS in the wired systems. Such evolutions are the leading driving forces for network operators to move towards the implementation and integration of NGN in their platforms.

Although NGN brings new features to telecommunication services and networks, it faces a number of inadequacies. In this chapter, we focus on the service interaction management issue as the main hurdle towards service orchestration in IMS. Service orchestration is a means for controlling the consecutive interaction between services. Even if a number of functionalities for the session establishment in IMS are well defined, there are still major challenges for supporting service orchestration. These challenges are twofold: the management of negative interactions, which end in the conflicts between services, and the management of positive interactions, which enable the composition of services.

The contribution of this chapter is to investigate the service conflict and service composition problems in IMS by discussing the corresponding requirements, principles, and challenges. In the next section, we present the need for a service conflict management mechanism in IMS and we review the related research work dealing with this issue. Based on the shortcomings of these propositions, we expose the key guidelines for designing a service conflict management mechanism in IMS. Likewise, in the "Service Composition Management in IMS" section, we explore the service composition management issue. After providing insight into the related work, the limitations of these solutions are discussed and the main aspects of a service composition manager in IMS are cited. In the conclusion to this chapter, we discuss the perspectives of the developed subjects and propose several trails for the future research work in the service architecture domain of NGN to follow.

Service Conflict Management in IMS

The IP telephony services may face the *service conflict* issue that was met in the traditional telephony networks, called the *feature interaction* problem. This problem occurs when services (or features) invoked during a session behave correctly when processed separately and independently of each other, but

not when running together. Here, by "feature" we refer to an independent service unit that a network provides to its subscribers in order to enrich and personalize the offered services.

Call barring, call forwarding unconditional, and originating call screening are among the well-known call control features with the following definitions:

Call barring restricts all outgoing calls to the network-defined forbidden destinations.

Call forwarding unconditional forwards all incoming calls to another user-defined destination.

Originating call screening blocks all outgoing calls to the destinations that are on the blacklist of the caller.

Although much research work has explored managing the service conflicts, applying these solutions in IMS remains challenging and further investigation is needed.

Service Conflict Example

In the following scenario, illustrated in Figure 14.1, a service conflict occurs between the services of the caller and the callee:

> Suppose that Alice wants to establish a video call with Bob. Alice has an originating call screening service that blocks all the outgoing calls to end users who are on the blacklist. On the other hand, Bob has a call forwarding unconditional service that sends all the incoming calls to Anne. What happens if Anne is on the blacklist of Alice? In this case, the following conflict occurs between the originating call screening service of Alice and the call forwarding unconditional service of Bob: If the originating call screening service works correctly as defined for Alice, then the call forwarding unconditional service will be neglected. Otherwise, if the call forwarding unconditional service behaves as it is defined for Bob, then the originating call screening service of Alice will be ignored.

Service Conflict Management

Managing service conflicts consists of *preventing* them whenever possible or of *detecting* and *resolving* them. Service conflict prevention is achieved by enforcing necessary constraints on the service platform and users. These constraints mainly deal with odd service configurations, inaccuracy in service features, bad resource allocations, software application programming errors, and the lack of interoperability between heterogeneous telecommunication systems. However, even if service platforms and end users respect these constraints, service conflicts are likely to occur.

In the literature, two methods are considered for detecting and resolving service conflicts: offline and online. A number of service conflicts can

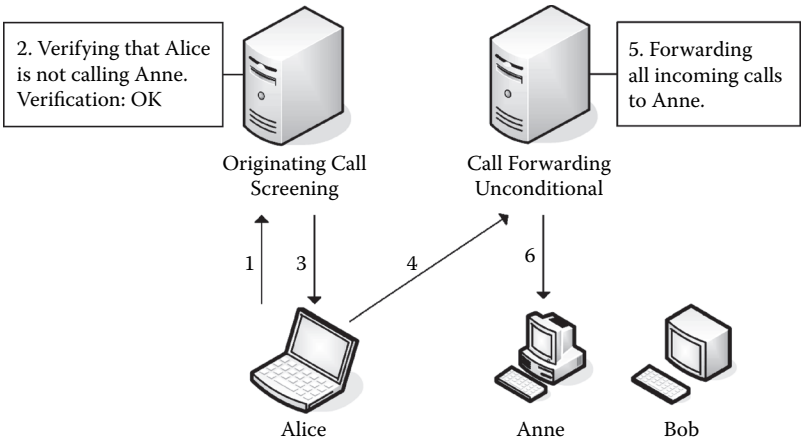

FIGURE 14.1
Service conflict example.

be detected and resolved before services are invoked. This management mechanism is called offline. However, due to the unpredictable behavior of services as well as the vast introduction of new services, all service conflicts cannot be detected and resolved offline. In other words, some conflicts occur during service run-time and consequently must be handled at service run-time—that is, online. Furthermore, we refer to a service conflict detection and resolution mechanism as "static" if it is based on the predictable and predefined service information. On the other hand, the management mechanisms based on real-time service information are called "dynamic." In the following sections, we present an inclusive review of the related work dealing with this issue.

Related Work

Surveys in Keck and Kuehn [3] and Calder et al. [4] give a comprehensive overview of the work carried out in this field during the 1990s. The work in Cameron et al. [5] classifies the causes of the feature interactions (violation of the feature assumptions, limitations on the network support, and problems in the distributed systems such as timing and race conditions) and presents two general approaches to manage the service conflicts: (1) avoiding the conflicts by improving the service architecture, and (2) providing the service architecture with specific service conflict detection and resolution methods.

Service Conflict Avoidance: Service Negotiation

Kolberg and Magill [6] propose a SIP-based service conflict avoidance mechanism in which end users negotiate their available services before the ses-

sion is established. In order to achieve this negotiation, the caller extends the INVITE request, addressed to the callee, by adding to the request a list of its services as well as the possible actions that the callee could perform if the latter does not agree with the indicated service list. Afterwards, the callee may accept the call or may select one of the offered actions. It is also possible that the callee refines some of the proposed actions and suggests the new ones to the caller. Then, the caller will decide which of the options is preferable. This negotiation keeps on until the caller and the callee agree on services and their relevant options.

This mechanism ensures that the session established between end users will not encounter the service conflict problem. However, the main shortcoming of this proposition is the lack of transparency for the end parties because they must be engaged in the service conflict management procedure. Moreover, the end parties should trust each other during the negotiation procedure; otherwise, the negotiation is vulnerable to many attacks, such as identity usurpation or denial of service (a "man in the middle" may answer the invitation request with an invalid service list, leading to the failure of the service negotiation). Finally, this proposition requires that the end parties agree on a service negotiation template in order to understand the negotiation vocabulary used.

Service Conflict Avoidance: Service Description

Harada, Fujiwara, and Ohta [7] propose a service conflict avoidance method based on predefined service conflict tables that contain a description of services. These service specifications are bound to terminals. Service conflicts may therefore occur due to combinations of two service specifications. The combinations are stored in tables that use state transaction diagrams for describing the potential service conflicts as well as the avoidance mechanisms.

The experimental implementation and evaluation of this proposition confirm its effectiveness in terms of conflict avoidance process time. But this proposition may not be flexible when new services are introduced. In fact, the service conflict tables should be regularly updated and bringing these tables up to date requires additional efforts by the proposed service conflict avoidance method.

Service Conflict Avoidance: Formal Models

Chan and Bochmann [8] provide a formal model of SIP services and service conflicts, using specification and design language (SDL) that covers some of the important characteristics of SIP messages, such as caller-ID, addressing header fields, and sequence number. In this proposition, the following service conflict prevention strategies are discussed: preventing resource contention and limitation, preventing incoherent interactions, preventing unexpected nondeterminism and unfair interactions, and preventing deadlocking interactions (due to the mutual deployment of services).

By means of the proposed service conflict classifications and the corresponding conflict prevention strategies, the number of potential service conflicts is supposed to decrease. However, these propositions face limitations regarding the quantification of service conflict instances and their behaviors. Furthermore, the proposed models must be regularly updated in order to manage the conflicts that may occur once new services are introduced.

Service Conflict Detection and Resolution: Offline

Chentouf, Cherkaoui, and Khoumsi [9] propose a service conflict detection and resolution method that integrates a feature interaction manager (FIM) agent between the end users and the SIP proxies. In this proposition, users inform the FIM about their intention to run a service. Then, the FIM launches the detection procedure to verify if this service will interact with the already running ones that belong to the same session. If a conflict is detected, then the FIM instructs the proxy to stop the call processing and launches the resolution procedure.

Khoumsi [10] presents a formal model based on finite state machines for specifying the services and a methodology to detect and resolve the service conflict based on predefined conflict cases. In this proposition, the service conflicts are defined as undesirable states of a finite state machine. As for the resolution method, this model forces the undesirable states to behave in an acceptable way.

Jouve, Gall, and Coudert [11] present the telecommunication services in the form of operational specifications and define a method to compute service triggering and the potential conflicts. Furthermore, this work proposes a mechanism to adapt these service specifications and conflict detection methods to SIP-based services.

Crespo, Carvalho, and Logrippo [12] propose a distributed conflict resolution mechanism for Internet applications. In this proposition, an interaction resolver advisor is designed that contains the list of services and the potential conflicts that may occur between them. When an application identifies several services that may be executed simultaneously, the list of these services is sent to the advisor. This later selects the services to be executed on the basis of policies expressed by sets of logical formulas (through the *Advice* response) and sends it back to the application.

Kolberg and Magill [13] propose an offline approach for describing the services and defining the rules to detect the service conflict scenarios. The detection procedure consists of comparing the specifications of one service with all the variations of the specification of another service. Kolberg and Magill [14] follow up this approach and propose a distributed service conflict detection and resolution mechanism that is applicable to SIP-based services. In this proposition, the behavior of services is describer via an extended SIP header. The triggered service includes this SIP header in the SIP message. Afterwards, the next invoked service compares the pairs of the invoked services to find out the conflicts. This comparison is based on the defined con-

flict detection rules. Once a conflict is detected, the predefined resolution algorithm decides about disabling one service in order to resolve the conflict that occurred.

The main drawback of these propositions is that they are limited to offline service conflict management solutions, and the conflicts occurring at service run-time could not be handled.

IMS Requirements for Service Conflict Management

To the best of our knowledge, not enough research on the service conflict management in IMS has been conducted so far. However, as a leading all-IP service control plane, IMS needs further consideration in this domain. Based on the lessons learned from the previous work, we define the following requirements for introducing a service conflict management mechanism in IMS:

- *Adaptability to IMS via SIP-based mechanisms*. Because service conflicts occur during the service invocation in IMS and because the session establishment and service invocation mechanisms of IMS are SIP based, any solution for service conflicts must be SIP based. In addition, the SIP-based nature of such a solution will provide it with extensible mechanisms to integrate new services and the relevant service conflict management aspects.
- *Transparency to the end parties*. Instead of providing the end parties with mechanisms to deal with the service conflict issue, this mechanism should be included in the core of IMS. The advantage of this approach is mainly related to the flexibility of IMS to functional modifications without engaging the end parties in this procedure.
- *Efficiency without considerable impact on the session establishment time*. As the service conflict management mechanism intervenes during the session establishment phase, it must respect the real-time constraints of the SIP sessions.

In order to fulfill these requirements, as illustrated in Figure 14.2, the following functional enhancements should be provided in the IMS standards:

- Introduce a functional entity to orchestrate the service invocations and to detect and resolve the service conflicts. This functional entity can be defined as a service broker that intercepts the messages exchanged between an S-CSCF in the session control layer and an application server in the service layer.
- Define a SIP-based service conflict detection and resolution procedures in the service broker, which, based on static and/or dynamic information, fits the defined requirements and handles both offline and online service conflicts.

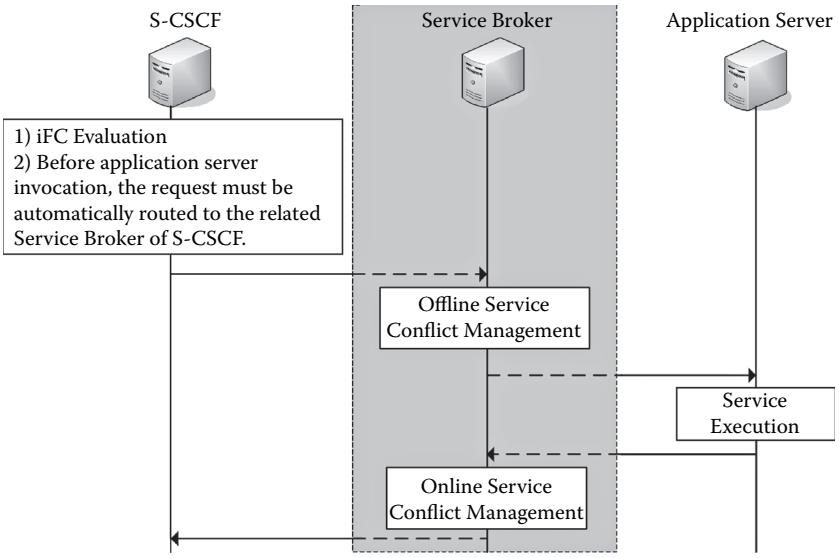

FIGURE 14.2
Service broker for managing service conflicts in IMS.

A standardization effort is ongoing at 3GPP for defining the architecture of the service broker and the associated procedures [15]. However, the progress of this work item has been slow and the discussions are still at a very high level. Only the basic principles will be standardized in 3GPP release 8; most of the specification work will be done in release 9.

Service Composition Management in IMS

The 3GPP and the NGN working group of TISPAN follow a modular approach for defining the independent and self-contained service building blocks, called *service capabilities*. Thanks to this modular approach, IMS provides the service capabilities to be used independently or reused by different application servers. Enabling such interoperability and cooperation between different services allows enriching and personalizing the application servers based on the user needs, preferences, and contexts provided by the service capabilities. Furthermore, this cooperation prevents the overlap and redundancy of the same functionalities offered by service capabilities and implemented by different application servers. Finally, the composition mechanism reduces the time-to-market of services. This concept should lead to easier and low-cost integration and deployment of application servers. Supporting this service level interworking by IMS is particularly interesting for the

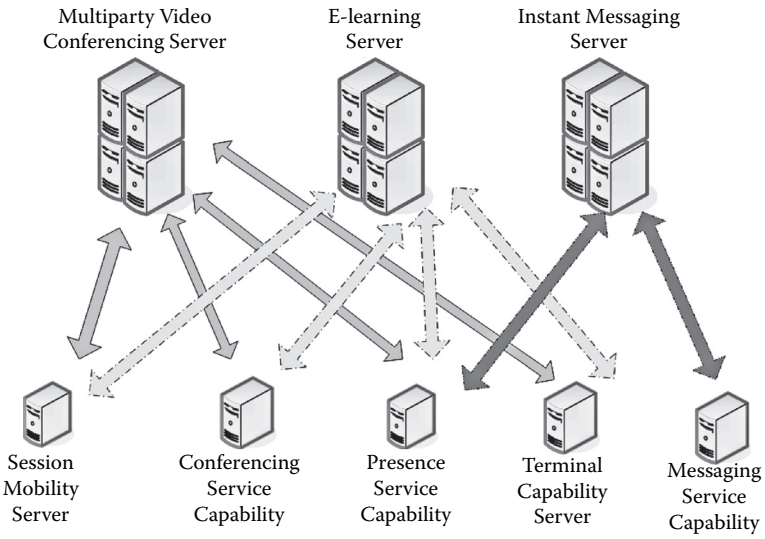

FIGURE 14.3
Service composition example.

third-party service providers, who can offer new application servers composed of IMS service capabilities.

In this section, after presenting several service composition examples, we outline the related works to deal with the service composition management issue. Then we describe the challenges to introduce an IMS-compliant service composition management solution.

Service Composition Examples

Figure 14.3 illustrates a few service capabilities and the application servers that can share and reuse them. For instance, an e-learning application server uses the presence service capability [16] to manage the availability of participants in an e-learning class. Furthermore, it uses a conferencing service capability [17] for setting up the conversation between them. Also, the e-learning participants are capable of transferring the multimedia session from one terminal to another terminal (e.g., from a mobile phone to a PC) through the session mobility service capability whenever they want to do so. The terminal capability server can cooperate with an e-learning application server to ensure the compatibility of the context to be sent to the end users based on the capability of their terminals. A multiparty video conferencing application server can use the same service capabilities to set up a conference based on the availability of the participants. In addition, an instant messaging application server uses the presence service capability to transfer messages between users based on their presence status.

Supporting such cooperation between services will appeal for mechanisms that tailor service interactions and manage the composition of services.

Service Composition Management

As an open service control platform, IMS is a step towards network convergence while trying to support the successful cooperation between application servers and service components across different networks. Indeed, by means of service composition, all kinds of service interworking scenarios can be achieved independently of the underlying control functionalities. Managing the composition of services mainly consists of *making available the service capabilities for application servers*, *controlling the access of different application servers to service capabilities*, and *managing the interactions between application servers and service capabilities by respecting the confidentiality of user context*. In addition, it must be noted that the effectiveness of a composite application server is highly dependent not only on the user preferences, but also on the context (network conditions and constraints) in which the user will access the application server. As a result, this will enable the application servers to be adaptable and easily reconfigurable to both user and network preferences.

Related Work

In the literature and in the standards, several solutions are specified for achieving service composition. Next, we review these solutions and discuss their capacity for realizing service interworking in IMS.

OMA-Based Service Composition Management Solution

The Open Mobile Alliance (OMA) [18] is the focal point for the development of mobile service entities (called *enablers*) and ensures the seamless service interoperability across different devices, service providers, and network operators. The cooperation between OMA and 3GPP enables OMA to leverage the IMS capabilities in an interoperable way by specifying the emerging service enablers, including the *instant messaging and presence service (IMPS) enabler* [19], *push-to-talk over cellular (PoC) service enabler* [20], and *OMA gaming service enabler* [21]. In these service examples, OMA applies application programming interfaces (APIs) for managing the composition of service enablers and capabilities that are inherent to IMS, such as presence and messaging [22] service capabilities.

However, the cooperation between services is only limited to the OMA enablers. Also, the specifications do not consider how to enable the share and reuse of service capabilities by different application servers. For instance, OMA does not specify how the presence service capability that is used in OMA IMPS can be shared and reused by the OMA PoC or by OMA gaming service enablers. Finally, supporting a multiprotocol environment by OMA

in order to remain compliant with the SIP-based nature of IMS introduces additional protocol adaptation features.

OSA/Parlay-Based Service Composition Management Solution

The Parlay Group [23] is a multivendor consortium founded to develop open, technology-independent APIs that enable the development of applications operating across multiple networking-platform environments. Parlay is protocol agnostic and aims to abstract the network capabilities into a set of APIs that allow services and applications to access the core network functionality transparently.

Even if the OSA/Parlay architecture aims to enable the integration of service capabilities specified by different standardization bodies, it cannot be directly applied as a means for service composition management in IMS. In fact, in order to adapt OSA/Parlay architecture to IMS, a Parlay gateway must be used on top of IMS components, through Parlay API. In this case, the gateway must translate the Parlay API methods to the SIP messages. Furthermore, the functionalities defined over the Parlay gateway cover only the service access control and service discovery issues. The composition management of services and the way they should share common service building blocks are not specified. Thus, an open service convergence environment is necessary to use Parlay-based services in a distributed network environment. Finally, the OSA/Parlay-based solution implies a heavy dependency of services on those defined in OSA/Parlay standards.

Ontology-Based Service Composition Management Solutions

Another research direction focuses on ontology-based mechanisms to enable the following features: *service description, service discovery, service invocation,* and *service composition and interoperation.*

Yokohata et al. [24] propose a service composition management method usable in mobile Internet environments. This method uses ontology Web language for services (OWL-S) to model service scenarios and to define a sequence of service elements to be invoked. OWL-S ontology is composed of three main parts: service profile (describes the service), process model (presents the interaction constraints with this service), and service grounding (details the interaction means with this service). Once services are modeled, a service composition engine translates the scenario, finds appropriate service elements (based on the semantics and the situations or contexts of the user), and invokes the service elements through Web service interfaces.

Another ontology-based service composition mechanism is proposed in Tarkoma et al. [25]. It uses the ontology and semantic Web technologies for integrated knowledge management in mobile service platforms. This proposition supports dynamic adaptability to user requirements. Furthermore, the automatic interaction of mobile services is ensured.

However, the service models represented by these ontology-based mechanisms have to cope with limitations in terms of expressing the definition of services and their needs and behaviors. In addition, the proposed service composition engine must be regularly updated in order to ensure the interoperability of the new services with the service platform.

Service Capability Interaction Manager (SCIM)

3GPP introduced SCIM [26] as a function within the SIP application server domain of IMS for managing the interactions between application servers. However, the functionalities of SCIM are not specified and research in this field is in progress.

Work described in reference 27 presents different ways to design SCIM, such as:

- a request dispatcher within the local execution environment for service logics;
- an interaction manager between the components that implement SIP proxies or user agents;
- an interaction manager between service capabilities that are exposed using Web service definition language (WSDL)- and simple object access protocol (SOAP)-based abstractions of IMS; and
- an interaction manager between SIP features and legacy signaling system components.

Choosing among these different behaviors of SCIM depends on the technological and business model issues of the different service providers and is still regarded as an open subject.

Araki, Yamamoto, and Sweeney [28] propose a group entertainment service platform that deals with the provisioning requirements of mobile entertainment services and provides them with a dynamic service composition mechanism. SCIM is used in this group entertainment service platform with the following functionalities: user context setting/retrieving, service provisioning, and service invoking. However, the SCIM defined in this proposition must be extended in order to cover the interoperability between service components.

To overcome the limitations of these solutions, in the following section we present the requirements for providing IMS with a service composition management mechanism.

IMS Requirements for Service Composition Management

Realizing the cooperation between services necessitates a service composition mechanism in IMS. This mechanism implies the following features:

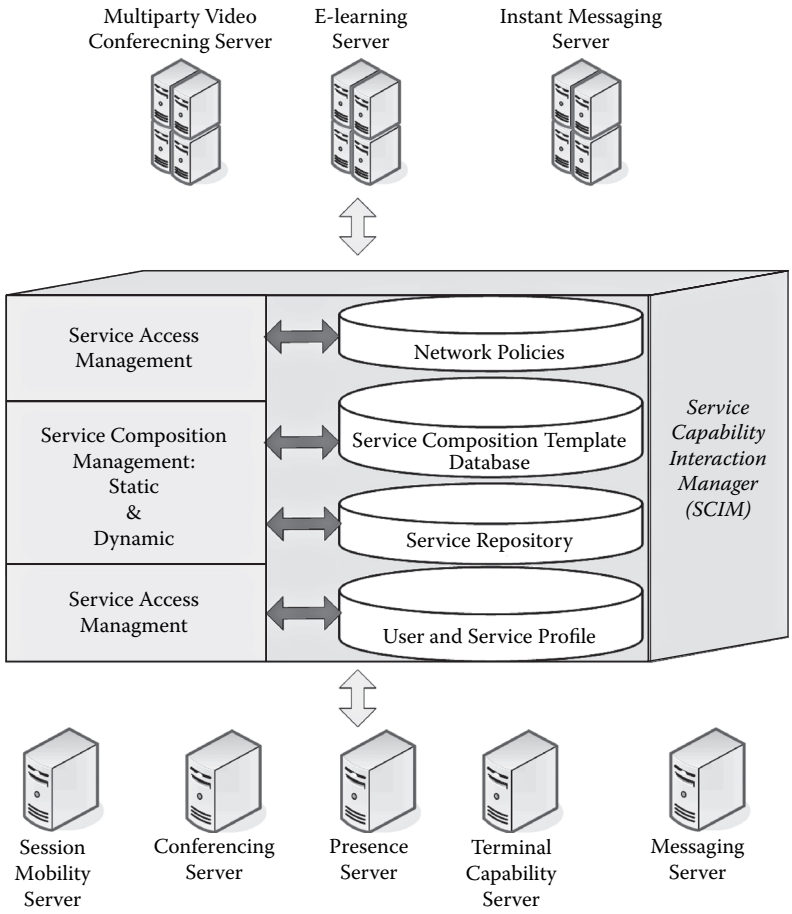

FIGURE 14.4
SCIM for managing service composition in IMS.

- The service composition management solution for IMS must respect the open architecture of IMS, where different services in different networks are flexibly supported.
- This mechanism must be adaptable to IMS. This adaptability can be achieved by using SIP-compliant service interoperability management approaches.
- This solution must be extensible to the introduction of innovative and personalized services.

To achieve these goals, as illustrated in Figure 14.4, SCIM could be used coupled with a service composition management algorithm. The two main functions of this algorithm include:

- *Service access management*. This function controls the access of the application servers to the service capabilities. This control consists of authenticating the application servers (through the network policies and the service agreements between network operators and service providers) and authorizing the service access. Such control is necessary, particularly when a third-party service wants to use a service capability of the network.
- *Service composition management*: This function controls the reuse of the service capabilities by the application servers. This control can be static (i.e., based on a predefined service composition template indicating which application server needs which service capability in which condition). For example, an e-learning application server should be composed of a presence server, a messaging server, and a conferencing server. In this example, the successive set of service capabilities that will be used by the e-learning application server will be presented statically in a service composition template. However, not all the integrated services can define their composition statically. For instance, depending on the contextual environment, a multimedia conferencing server may contact the terminal capability server in order to send the appropriate media to the users according to the characteristics of their terminals. Hence, the cooperation between the multimedia conference server and the terminal capability server depends on the user preferences and can hardly be static. Therefore, in this example, the terminal capability server must be dynamically (and on user demand) available for the multimedia conferencing server. Consequently, a dynamic service composition management mechanism is required.

Conclusion

NGN has emerged in the past couple of years as a means for providing services to users in heterogeneous networks. Alongside these innovations, as the leading NGN service control overlay, IMS is specified to handle the delivery of multimedia services. However, the current IMS specifications go through critical shortcomings when interacting with services. Such service interaction may be negative or positive. Negative service interactions result in conflicts between services and end to unexpected behaviors of services. Positive interactions, on the other hand, enable the creation of enriched services that are composed of modular service building blocks.

AU: "an end to unexpected..." or "end in unexpected..."?

In the light of IMS growth towards a unique service control platform, the lack of service interaction management is unacceptable. The analysis developed in this chapter on the NGN reference model's standards, mainly on

IMS specifications, pointed out the inadequacy of the standards for handling service interactions. Based on an inclusive survey on the related research work, the requirements and principles for introducing an intelligent service orchestration mechanism in IMS are defined.

Because IMS is an evolving standard and NGN services witness increasing evolutions and innovations, there are many perspectives in the IMS service interaction management direction. Implementing the service broker and SCIM over the IMS platform may be the first step.

References

1. 3GPP, TS 23.2281. IP multimedia subsystem (IMS), 3GPP TS 23.228.
2. Rosenberg, J. et al. 2002. SIP: Session initiation protocol, IETF RFC 3261.
3. Keck, D. O., and P. J. Kuehn. 1998. The feature and service interaction problem in telecommunications systems: A survey. *IEEE Transactions on Software Engineering* 24(10):779–796.
4. Calder, M., M. Kolberg, E. H. Magill, and S. Reiff-Marganiec. 2003. Feature interaction: A critical review and considered forecast. *The International Journal of Computer and Telecommunications Networking* 41(1):115–141.
5. Cameron, E. J., N. D. Griffeth, Y.-J. Lin, M. E. Nilson, W. K. Schnure, and H. Velthuijsen. 1993. A feature-interaction benchmark for IN and beyond. *IEEE Communication Magazine* 31(3):64–69.
6. Kolberg, M., and E. H. Magill. 2001. Handling incompatibilities between services deployed on IP-based networks. *IEEE Intelligent Networks* 2001:360–370.
7. Harada, D., H. Fujiwara, and T. Ohta. 2006. Avoidance of feature interactions at run-time. *IEEE International Conference on Software Engineering Advances*:6–12.
8. Chan, K. Y., and G. V. Bochmann. 2003. Methods for designing SIP services in SDL with fewer feature interactions. In *Feature interaction in telecommunications and software systems VII*, 59–77. Amsterdam: IOS Press.
9. Chentouf, Z., S. Cherkaoui, and A. Khoumsi. 2003. Implementing online feature interaction detection in SIP environment: Early results. *IEEE International Conference on Telecommunications* 2003:515–521.
10. Khoumsi, A. 1997. Detection and resolution of interactions between services of telephone networks. In *Feature interaction in telecommunications networks IV*, 78–92. Amsterdam: IOS Press
11. Jouve, H., P. L. Gall, and S. Coudert. 2005. An automatic offline feature interaction detection method by static analysis of specifications. In *Feature interactions in telecommunications and software systems VIII*, 131–146. Amsterdam: IOS Press.
12. Crespo, R. G., M. Carvalho, and L. Logrippo. 2006. Distributed resolution of feature interactions for internet applications. *Elsevier Computer Networks*:382–397.
13. Kolberg, M., and E. H. Magill. 2002. A pragmatic approach to service interaction filtering between call control services. *The International Journal of Computer and Telecommunications Networking* 38(5):591–602.
14. Kolberg, M., and E. H. Magill. 2007. Managing feature interactions between distributed SIP call control services. *Elsevier Computer Networks* 51(2):536–557.

AU: pls provide vol #

15. 3GPP, TR 23.810. Study on architecture impacts of service brokering.
16. 3GPP, TS 24.141. Presence service using the IP multimedia (IM) core network (CN) subsystem, 3GPP TS 24.141.
17. 3GPP, TS 24.147. Conferencing using IP multimedia (IM) core network (CN) subsystem, 3GPP TS 24.147.
18. OMA. Available from http://www.openmobilealliance.org
19. OMA-IMPS. Enabler release definition for IMPS, V1.3, OMA-ERELD-IMPS-V1_3-20070123-A.
20. OMA-PoC. Enabler release definition for push-to-talk over cellular, V 1.0.1, OMA-ERELD-PoC-V1_0_1-20061128-A.
21. OMA-Games. Enabler release definition for game services, V 1.0, OMA-ERELD-Games-Services-V1_0-20030612-C.
22. 3GPP, 23.140.
23. Parlay. Available from http://www.parlay.org
24. Yokohata, Y., Y. Yamato, M. Takemoto, and H. Sunaga. 2006. Service composition architecture for programmability and flexibility in ubiquitous communication networks. *IEEE International Symposium on Applications and the Internet Workshops.*
25. Tarkoma, S., C. Prehofer, A. Zhdanova, K. Moessner, and E. Kovacas. 2007. SPICE: Evolving IMS to next-generation service platforms. *IEEE International Symposium on Applications and the Internet Workshops.*
26. 3GPP, TS 23.002. Network architecture.
27. dev2dev. Service Capability Interaction Management (SCIM) in IMS. http://dev2dev.bea.com/blog/mpalmete/archive/2006/02/service_capabil.html
28. Araki, Y., A. Yamamoto, and M. Sweeney. 2007. Dynamic community entertainment service composition on next-generation mobile network IP multimedia subsystem. *IEEE International Symposium on Applications and the Internet Workshops.*
29. Cortez, M., J. R. Ensor, and J. O. Esteban. 2004. On SIP Performance. *Bell Labs Technical Journal* 2004:155–172.
30. Crespi, N. 2006. A distributed mechanism to resolve dynamically feature interaction in the UMTS IP multimedia subsystem. *International Workshop on Applications and Services in Wireless Networks.*

AU: pls provide complete ref

AU: refs 29 & 30 not cited in text; OK to delete?