

University of East London Institutional Repository: <http://roar.uel.ac.uk>

This paper is made available online in accordance with publisher policies. Please scroll down to view the document itself. Please refer to the repository record for this item and our policy information available from the repository home page for further information.

To see the final version of this paper please visit the publisher's website. Access to the published version may require a subscription.

Author(s): Falcarin, Paolo; Lago, Patricia.

Article title: Call Control Component implementing converged Telephony- Internet networks

Year of publication: 2001

Citation: Falcarin, P. and Lago, P. (2001) 'Call Control Component implementing converged Telephony- Internet networks', In: Proceedings of the 3rd International Workshop on Net-Centric Computing, Toronto, Canada, May 2001, pp. 32-35.

Link to published version: <http://netcentriccomputing.org/2001/proceedings.htm>

Call Control Component implementing converged Telephony-Internet networks

Paolo Falcarin

Dipartimento di Automatica e Informatica
Politecnico di Torino
Corso Duca degli Abruzzi 24
I-10129 Torino, Italy
+39 011564.7091
Paolo.Falcarin@polito.it

Patricia Lago

Dipartimento di Automatica e Informatica
Politecnico di Torino
Corso Duca degli Abruzzi 24
I-10129 Torino, Italy
+39 011564.7008
Patricia.Lago@polito.it

ABSTRACT

Global telecommunications are in the early stages of a fundamental transformation, in which telecommunication networks and information technologies are yielding converged networks named Next Generation Networks (NGN). In this context this article presents a call control component able to handle calls between a telephone and a PC, accordingly to actions decided by the application layer that sets up and redirects calls between terminals over heterogeneous networks like Internet and PSTN. This goal is achieved through software adapters offering APIs porting standard protocols like Media Gateway Control Protocol and Intelligent Network Application Protocol (INAP), to CORBA-based software components.

Keywords

Call Control, MeGaCo, SIP, INAP, Parlay, Media gateway.

1 INTRODUCTION

The greatest benefit of the new trend of telecommunication systems is the openness of interfaces and protocols that permits interaction among components implemented by different vendors or service providers. In fact, past telecommunication scenarios were based on e.g. traditional circuit switching that bundles proprietary solutions and is generally providing voice-centric services. On the contrary, Next Generation Networks (NGNs) offer a decoupled switching architecture, with multi-service support over a common media-transport layer, lower cost implementation, and reuse of 3rd party components.

Also, this impulse in telecommunications is producing a rapid change within telecommunication services. Customers' requests for services simple-to-use, permitting to communicate through any kind of terminal, and at the same time customizable, is always increasing.

To permit communication *anywhere* in a transparent way, we developed an architecture (and in particular a set of components) that operate communication over hybrid converged networks (in particular PSTN and Internet), and that carry service logic to customize this communication routing according to customer needs or provider network resources. These components (supporting Call Control) are based on distributed object-oriented technologies, and encapsulate the logic to treat converged networks as legacy systems, and provide standard interfaces for multi-provider inter-operability.

2 OBJECTIVES AND CONTEXT

This work focused on the development of a call control component able to establish, manage and finalize calls among a telephone and a PC, according to the actions "decided" by the application layer. The call model has to manage in a homogeneous way the characteristics of the different communication protocols used by the involved physical network resources. In this respect, the implementation has been preceded by the analysis of different protocols for standard communication like INAP [3], SIP [9] and MeGaCo [5]. The chosen programming language was Java, and the implemented software components are: the Call Control server (CC) and the Virtual Presence¹ Call Control server (VPCC). The project notation is OMG/UML [6], extended with the definition of OMG/IDL interfaces. In tight collaboration with Telecom Italia Lab [10] researchers, we partially reused some interfaces defined in the Generic Call Control Service (GCCS) Parlay [8] specification and we extended related service diagrams and state diagrams to fulfill our architecture requirements. The internal architecture of the

¹ Virtual Presence [4] is a service implemented in the context of the European Eurescom Project P909 "Enabling Technologies for IN evolution and IN-Internet integration" [2]. It permits users to personalize the way they can be traced on hybrid converging networks, by use of User Profile.

two servers has been developed by considering also issues like concurrency management and real-time constraints, efficiency in the management of multiple concurrent calls, and deadlock-safety. At last, these components communicate through Orbacus, a CORBA [1] implementation made by OOC [7].

The components of the complete architecture are depicted in Figure 1. They can be classified in four categories: *network resources*, *adapters* acting as bridges between specific network resources and the architecture core, *services* made of Call Control components, and *applications*.

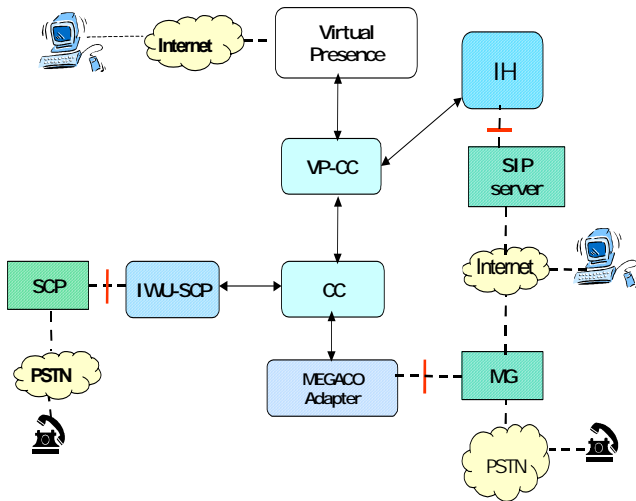


Figure 1: Architectural Overview

In particular, the components directly involved in the communication with the CC are: SCP and the Media Gateway (MG) with the respective adapters.

Service Control Point (SCP) manages real-time signaling in the Intelligent Network. Its behavior depends on its programmable service logic, which has been used in the Project to expand PSTN domain to IT, e.g. to ask an Internet application to resolve phone numbers in the PSTN-IN domain, according to customized preferences stored in the application layer. Exporting SCP features in our architecture is possible by using an adapter (see *IWU-SCP*) that bridges between SCP and the architecture, and that exports the commands and events of INAP protocol to IDL interfaces, implemented partly by this adapter and partly by CC.

Media Gateway (MG) converts media provided in one type of network to the format required in another type of network. E.g. a MG can terminate bearer channels from a switched circuit network and media streams from a packet network (e.g., RTP streams in an IP network). This gateway is capable of processing audio, video (alone or in

any combination) and full duplex media translations. A MG needs a Media Gateway Controller (i.e. the Call Control) to receive the correct commands, and send events notification. Commands and events adhere to the MEGACO/H.248 standard protocol that models a call as a *context* containing *terminations*. These represent sources and/or sinks of one or more media streams. Interaction between CC and MG is obtained through a MEGACO Adapter that exports a set of MEGACO-like IDL interfaces in order to interact with CC. The choice to specify an API founded on a protocol standard based on the possibility offered by this solution to make Call Control independent from the interfaces with the different protocols used by gateway providers.

On the VP-CC side, communication involves the *Invitation Handling component (IH)*. The IH manages communication requests (invitations) from/to the IP domain, by realizing a conversion among the signaling protocol messages to invite users to a service session, and the method invocations on the component interfaces inside the service layer. IH connects IP-terminals by means SIP, an application-layer protocol used for establishing and tearing down multimedia sessions, both unicast and multicast. It has been standardized within the Internet Engineering Task Force (IETF) for the invitation to multicast conferences and Internet telephone calls. A user is addressed by using an email-like address “user@host”, where “user” is a user name or phone number and “host” is a domain name or numerical address. Each new SIP transaction has a unique call identifier, which identifies the session. If the session needs to be modified, e.g. for adding another media, the same call identifier is used as in the initial request, in order to indicate that this is a modification of an existing session. IH has two basic functions: listening for in-coming SIP messages forwarded by a SIP proxy server, and sending SIP messages accordingly to user actions defined by the Application Layer. The SIP application on client-side also starts appropriate audio-conferencing applications according to the session that has been established.

3. CALL CONTROL

Call Control (CC) offers two kinds of functionality: a Parlay-like API that offers service interfaces for applications; multiple specific API towards network layer, in order to receive events notifications from different network resources, according to standard protocols used. CC also provides an abstract call model that can cover the essential features of all the call models used by the underlying network resources.

Virtual Presence Call Control (VPCC) has to be intended like an extension of the Virtual Presence application used to implement application-side Parlay interfaces and IH interfaces used to receive invitation messages from IP network through IH component. Inside the CC other two types of objects are used to model the concept of synchronous call between two terminals:

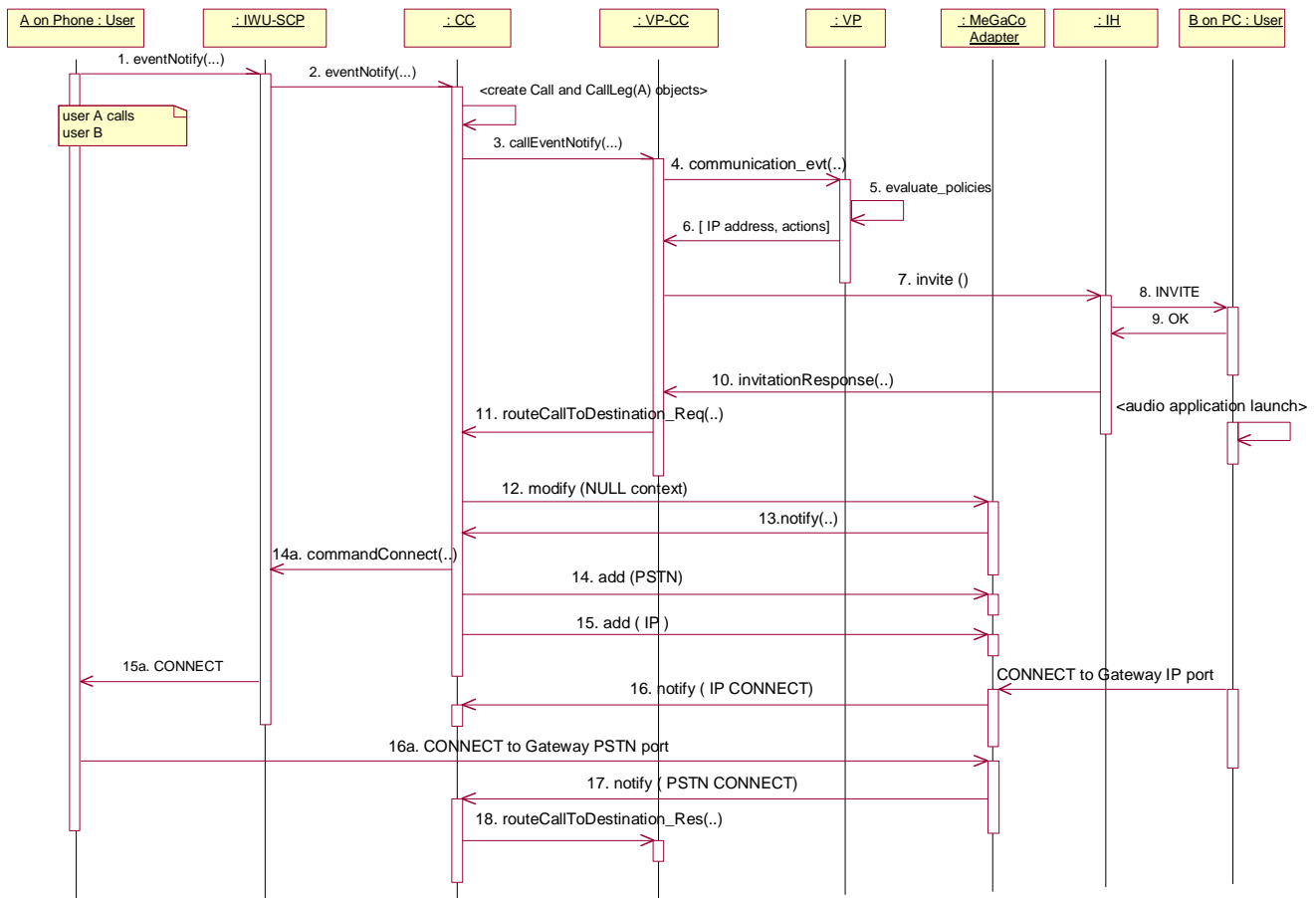


Figure 2. Sequence diagram scenario “Phone to PC”

- Class Call represents the information related to the execution of a call.
- Class Call-Leg represents all the information related to a terminal involved in the call.

Both classes are described by a state-diagram that reuse Parlay Call and Call-Leg states. Extension of these state diagrams have been produced after insertion of a new kind of transitions, represented by method invocations and parameters values described in MEGACO, INAP and IH interfaces implemented by CC and VPCC. Therefore CC and VPCC manage hybrid scenarios, because they set up calls between heterogeneous domains like PSTN and IP networks, e.g. both “Phone to PC” calls and “PC to Phone” calls. However CC and VPCC can handle classic homogeneous calls between two phones or between two PC connected by means of audio-conference applications. Another important feature is the uniform propagation of events notifications coming from different networks (like no answer, busy, disconnection, error) toward the application layer that can decide how to do depending by

call-handling user-defined policies: e.g. application can decide to redirect call to another terminal (phone or PC) or it can decide to close the call session. VPCC interprets actions decided by application and executes these commands interacting with CC. In figure 2 is represented the “Phone to PC” sequence diagram.

In Figure 2, user A composes the telephone number of a subscriber to the Virtual Presence service that will have the structure XXXX-ZZZZ, where XXXX represents the prefix that identify this service on national staircase (as the number 800 for free-phone) while ZZZZ is the Personal Number that it identifies the service subscriber. Through the SCP this information, together with the caller number, is passed to IWU-SCP and then forwarded to CC through the invocation of the method eventNotify () of its INAP-like. Now CC creates the object Call, assigning to it a unique identifier, and the object Call-Leg related to the caller and it transmits these data to the VP-CC invoking the method callEventNotify. VPCC invokes the method communication_evt () on the Virtual Presence that, after querying its private user database, decides how the

subscriber wants to handle this kind of calls.

The actions decided, and, in this case the destination IP address, are returned to VPCC that has to set up the hybrid call between a phone and a PC. It first of all invokes the method Invite() on the Invitation Handling interface, asking to it to send a SIP INVITE command (INVITE) to the IP address returned by the application layer. In the case that the user B accepts the invitation, the audio-conference application is launched on his PC and the message OK 200 is sent to the component of Invitation Handling. Now it will invoke the InvitationResponse() method on VPCC, to communicate to it that the called user has accepted the invitation to establish a synchronous communication. Now VPCC will ask CC to set up the hybrid call, invoking the method routeCallToDestination_Req() that triggers the creation of the second Call-Leg related to the called IP address and requires to the media-gateway to establish a Phone-to-PC call.

For first thing the CC invokes the method Modify () on the MEGACO interface of the Megaco Adapter in order to reserve a gateway phone-port for the arrival of a call from the net PSTN; this port number returned as a notify() parameter will be forwarded to the caller via the commandConnect () invocation on the INAP interface of IWU-SCP, in order to redirect the phone call to the gateway phone port. CC. With first Add(), CC commands to media gateway to listen to events of connection or disconnections from PSTN network. With second Add(), CC says to Media gateway to listen to events coming from IP network. Meanwhile the audio-conference application has already been performed, after invocation of invite() from Invitation Handling component, telling to called user's PC to connect with media gateway IP address. Therefore, after that both network connections have been set up, CC receives notify(IP-CONNECT) and notify(PSTN-CONNECT) invocations from Megaco Adapter. Since the gateway used by the implementation has a unique IP address, the IP termination allocated is always the same one and it is immediately passed to the called PC as an output parameter of invitationResponse() method. When two Call-Legs are both connected, CC invokes routeCallTo-Destiantion_Res() on VPCC interface to notify that call is active. VPCC has now to listen to the events of disconnection from the two connected terminals, and therefore it will release resources.

4. FUTURE WORK

This architecture can be enhanced in different ways: Call Control could be dynamically replied on several servers in order to support a major traffic load. Every CC could handle more than one Media Gateway, in order to obtain a more scalable architecture, able to reserve resources depending on traffic and QoS requirements.

Second, we are planning to include new network resources (e.g. supporting unified messaging) in order to handle

asynchronous communications too and to implement bi-directional translation of voice and text-based messages (like e-mail or SMS).

REFERENCES

- [1] CORBA Web site. On-line at <http://www.corba.org>
- [2] EURESCOM Web site. On line at <http://www.eurescom.de>
- [3] INAP protocol specification. On line at http://www.dfv.rwth-aachen.de/report/paragraph2_4_5_4_2_1.html
- [4] Lago, P., Licciardi, C.A., Canal, G., Andreetto, A., An architecture for IN-Internet hybrid services. In Computer Networks Journal, Special Issue on Intelligent Networks and Internet Convergence, T. Magedanz, H. Rudin, I. Akyildiz (Eds.), Elsevier, Vol. 35(5), April 2001, 537-549.
- [5] MEGACO protocol internet draft. On line at <http://www.ietf.org/html.charters/megaco-charter.html>
- [6] Object Management Group Web Site. On-line at <http://www.omg.org>
- [7] OOC Web site. On line at <http://www.ooc.com>
- [8] Parlay consortium Web site. On line at <http://ww.parlay.org>
- [9] SIP protocol Internet draft. <http://www.ietf.org/html.charters/sip-charter.html>
- [10] Telecom Italia Lab Web Site. On-line at <http://www.telecomitalialab.com/>.