

Extended Call Control Telecom Web Service

David E Vannucci & Hu E Hanrahan
Centre for Telecommunications Access and Services ¹
School of Electrical and Information Engineering
University of the Witwatersrand, Johannesburg, South Africa
Tel:+27117177229 Fax:+27114031929
{d.vannucci, h.hanrahan}@ee.wits.ac.za

Abstract—This paper investigates the requirements for a Parlay-X Extended Call Control (ECC) Web Service, and why it is necessary to support convergence of Internet and telecommunications. A Virtual Private Branch Exchange application is used to illustrate how advanced call control can be facilitated using an ECC Web Service. Using the example a methodology to develop the ECC methods is shown. A call model suitable for ECC is proposed.

Index Terms—OSA/Parlay, Extended Call Control, Web Service, Convergence, Call Model, Virtual PBX

I. THE CASE FOR WEB CALL CONTROL

In September 2005, Ericsson first identified a need for Web Service based advanced Call Control [1], due to a number of typical call control use cases not being realisable with the existing Call Control Web Services. A new specification “Extended/Advanced Call Control” was proposed, and subsequently adopted by the Parlay Group, as part of the API Acceleration programme (a programme which is successful in rapidly creating specifications). The Extended Call Control (ECC) Web Service aims to realise powerful stateful Web call control applications for Parlay-X by adding additional capabilities to standard call control, providing the ability to apply powerful capabilities, similar to that possible using Parlay interfaces.

Calls may be initiated by the application or network, and a Web Application can register control these calls via the ECC Web Service. In addition Web Service type abstraction is maintained [2]. ECC provides subscribed and triggered network level event notifications, long lived application control, control over network elements such as an IVR, access to network databases, and multiparty multimedia operations [2], [3].

Control via a Web service provides an IT developer telco functionality without having to be concerned with detailed understanding of equipment and network operation. An ECC Parlay X Web Service aims to provide loose coupling between the application and web service, high abstraction of operation, and still maintain flexible control over resources.

¹The Centre is supported by Telkom SA Limited, Siemens Telecommunications, Vodacom South Africa, Telsaf Data and the THRIP Programme of the Department of Trade and Industry.

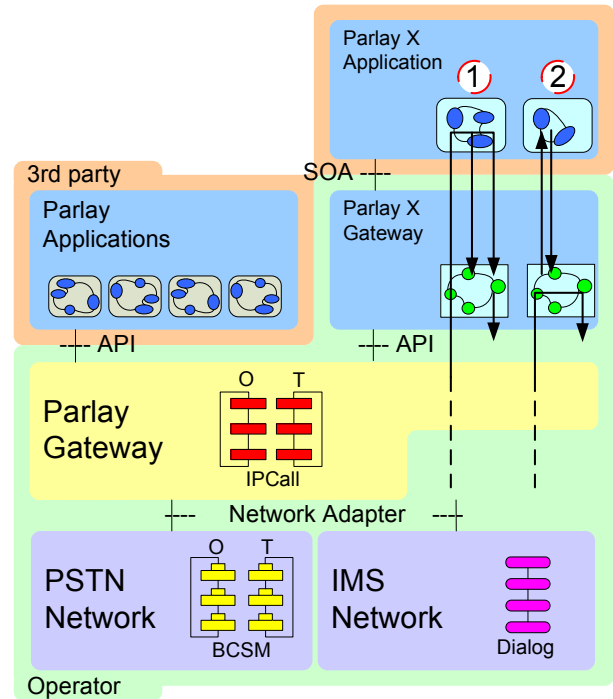


Fig. 1. Parlay X Architecture

The aim of Extended Call Control is rapid service development by developers having Web Service skills, and not the specialised skills required for IN programming, allowing the number of application developers to move into the millions [4].

Figure 1 shows a Parlay-X architecture and how an extended call application shown by the label (1) operates compared to a standard web service application shown underneath label (2). The ECC application operates over the entire duration of the call, receiving information as required from the ECC Web Service as to the state of the call and parties, and constantly provides required instructions to the ECC Web Service which is passed on to the network. Normal call control Web applications, as shown in figure 1, often provide only simple rule-based operation, such as creating a black/white list or a forward on busy, thus only being involved with the call for a very short part of its duration, or not at all.

In this paper we consider some of the requirements for an emerging telecommunications web ser-

vice, namely Extended Call Control as proposed by the Parlay Group. The intention of ECC is to allow greater cooperation between Internet developers and Telecommunications Service Providers, thus enabling the benefits of convergence. The Parlay-X Extended Call Control Web Service is as yet undefined; however there is a clear need for such a service. This paper provides an innovate method to develop an ECC Web Service in an extensible manner. A reference example is provided to illustrate how ECC can be used for a virtual private branch exchange application.

II. PARLAY AND PARLAY-X CALL CONTROL

Telecommunication networks are inherently complex, and to program service logic requires detailed knowledge of underlying protocols such as INAP, SS7, ISUP, and SIP [5]. The Parlay middleware aims to abstract network resources so that only an understanding of distributed software programming and telecommunication concepts is required, through the use of standardised Parlay interfaces. The OSA/Parlay gateway contains a number of network adapters, allowing the gateway to be connected to telco network equipment, as shown in figure 1. The Parlay gateway behaves like a telco Intelligent Network Service Control Point or IMS Application Server. The gateway analyses network information, and additional service logic is provided by Parlay applications resolves how to continue processing the call.

The telco network contains call models to represent the state of a call being processed in the network. When service logic is required, the Parlay gateway presents a notification to the Call Control Manager which creates appropriate objects to represent the network state. As shown in figure 1, call models occur with varying degrees of abstraction throughout the system layers.

The OSA/Parlay gateway interfaces are very rich in functionality but require a fundamental understanding of telecommunication call control, messaging and database operations [5]. Also, multiple method calls and callbacks must be understood. Parlay interfaces are thus thought to be too complicated for ease of use by IT developers. Parlay-X intends to provide simple call control. Current Parlay-X Call Control Web Services have, however, been found to be too simplistic. Extended Call Control is therefore intended to fill the gap between Parlay Application Call Control and current Parlay-X Call Control.

Through an ECC Web Service, fine grained call messaging and data operations are abstracted to the extent that these advanced operations can be provided in single method calls.

Parlay-X Call Control consists of five call control (CC) specifications (Third Party Call, Call Notification, Call Handling, Audio Call, Multimedia Conference).

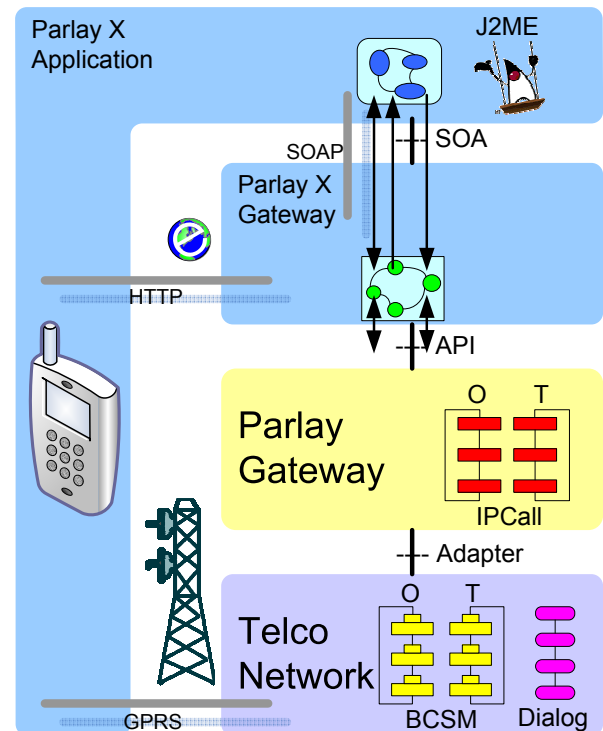


Fig. 2. Parlay-X Extended Call Control Application

These Web Services provide functionality which is difficult to combine to control more complex calls. Each Web Service is often designed with a particular application in mind, and any application that tries to operate outside of the intended use case very often cannot be implemented. Extensibility of current call control Web Services is also an issue due to non uniform data types.

III. EXTENDED CALL CONTROL APPLICATIONS

To demonstrate how an ECC Web Service could be used, we consider the example of a virtual receptionist's switchboard. This application would allow a company to operate in a completely distributed manner, with the receptionist having a Web application to control the company "extensions", which are standard mobile numbers. Using the Web application the receptionist is able to perform standard and advanced private branch exchange functionality. Due to all new mobile phones having packet data support, the Web application could be embedded on the mobile phone itself, allowing for application layer signalling as proposed in [6], as shown in figure 2.

A number is assigned by the operator to the switchboard, and when callers call in to the main number, the receptionist's mobile phone rings, and the Web based application also reports an incoming call. The receptionist then specifies which phone she wants to answer the call on, and the call is put though. Thus,

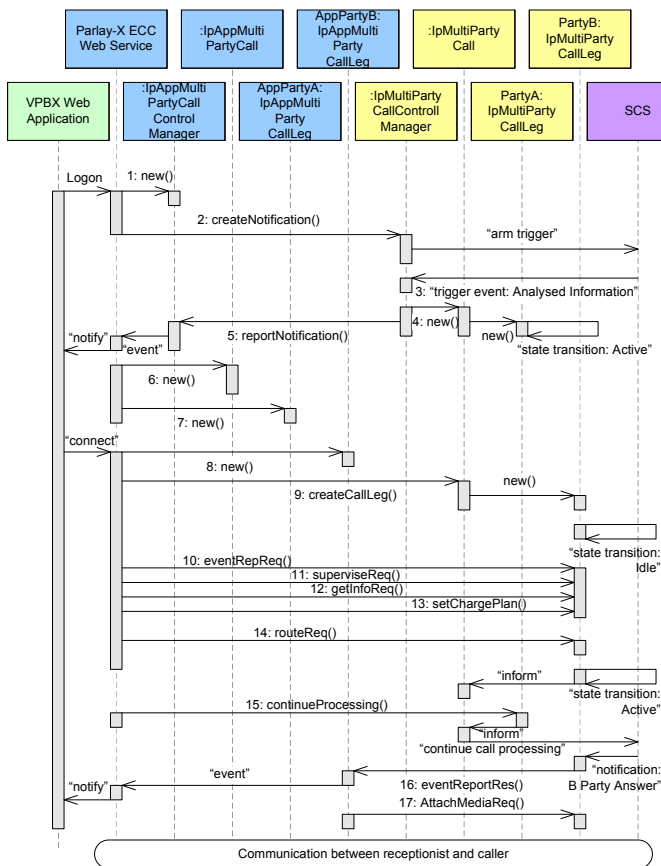


Fig. 3. Receptionist is alerted of incoming call and answers using mobile

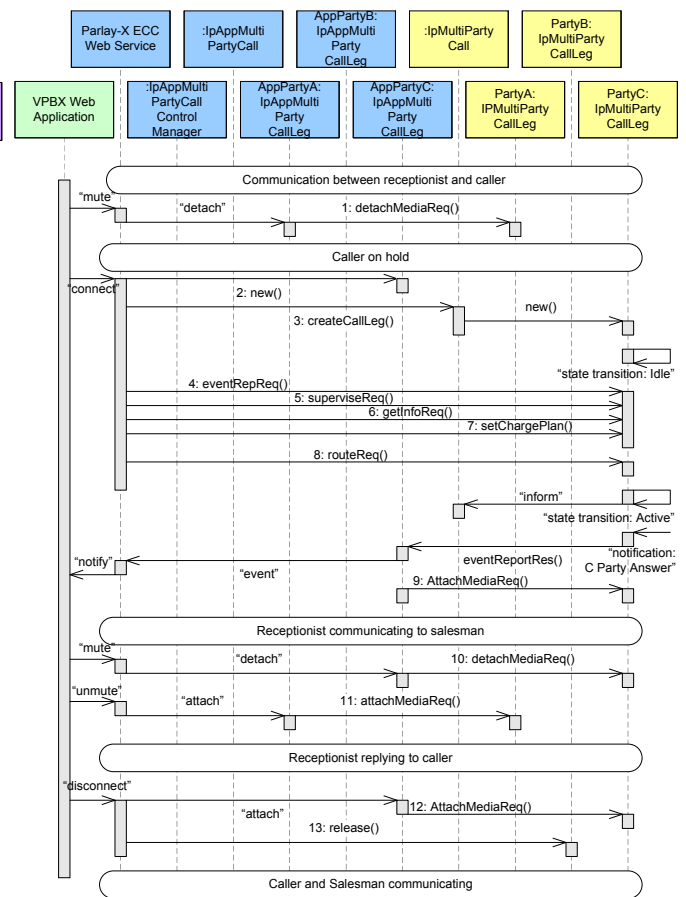


Fig. 4. Receptionist performs a conferenced transfer

the receptionist answers calls, and performs PBX functionality via the application. For the purpose of these examples this functionality would be:

- To be able to transfer the caller to a salesman’s mobile number, either first conferring with the salesman or doing a blind transfer. Requires: Call Leg Control, Call Creation.
- Create a multiparty call between the caller, their sales consultant, and technical support. Requires Call Leg Control and Multiparty Call Creation.

Thus, core functionality in an ECC Web Service is clearly the ability to manipulate call legs; this indicates that Parlay Generic Call Control is inadequate for advanced call control. In the following examples multi-party call control is considered to be the base Parlay capability function upon which to map ECC, since Generic Call Control does not support third-party initiated sessions, as shown in the inheritance of the Call Control APIs, figure 5 [7].

A. Answer the Call

In figure 3 the sequence of events to answer the call and connect it to the phone of choice is shown. All interfaces and methods are defined in the OSA/Parlay standard [9].

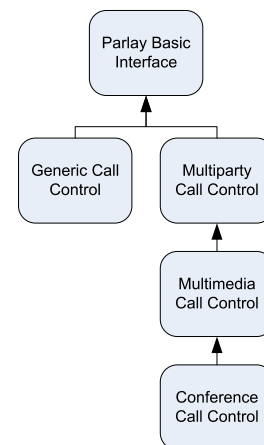


Fig. 5. OSA/Parlay Call Control Inheritance [8, pg. 163]

- 1: The Application creates an object implementing the `IpAppMultiPartyCallControlManager` interface.
- 2: This message enables the application to receive notifications of new call events to the company number.
- 3: When a new call to the company number arrives, a message (“analysed information”) is sent to the web service `IpMultiPartyCallControlManager`. Meeting the predefined criteria, messages are generated to create the call and call leg objects.
- 4: A new `IpMultiPartyCall` object, and `IpCallLeg` for the caller (Party A) is created to handle the call in

the Parlay gateway.

5: A notification is reported to the Web Service, call processing is suspended in the network whilst waiting for further instructions. References to the created objects are provided.

6,7: Using the provided reference, an application objects are created to manage the call and call leg.

8: An `IpAppCallLeg` is created to receive callbacks for the receptionist leg.

9: A new `IpCallLeg` object is created to the number the receptionist chooses to use (this can be predefined or changed according to location and receptionist preference). The call leg object is not yet routed in the network.

10: The Web Service requests to be notified of events such as when the receptionist answers the call, when the receptionist disconnects, or dials particular digits.

11: The Web Service requests to supervise the call leg to the receptionist, thus being in control of the call leg for its entire duration.

12: The Web Service requests information associated with the call leg, so that charging can be calculated.

13: The Web Service requests a specific charging plan for the call leg, as it is being used for the Virtual PBX service.

14: The Web Service requests to route the terminating receptionist leg to reach the receptionist, the call leg transits to the Active state.

15: The Web Service requests to resume call processing for the caller leg routing it to the designated receptionist phone.

16: When the receptionist answers the phone, the Web Service is alerted.

17: The media stream is attached to enable voice communication between the caller and receptionist.

B. Transfer the Call

In figure 4 the call has been established and the receptionist transfers the call to a salesman, first conferring with the salesman.

1: The media of the callers call leg is detached, preventing inband communication between the caller and other parties to the call.

2: An `IpAppCallLeg` is created to receive callbacks for the salesman's call leg.

3: A new `IpCallLeg` object is created to the predefined number of the salesman. The call leg object is not yet routed in the network.

4,5,6,7: The Web Service requests to be notified of events such as when the salesman answers the call. In addition the Web Service requests to supervise the call leg to the salesman, and receive information associated with the call leg, so that charging can be calculated. The specific Virtual PBX charging plan is applied.

8, 9: The Web Service requests to route the salesman leg. Once the call is answered the the Web Service is alerted, and the media stream is attached to allow communication between the receptionist and the salesman. The receptionist can now advise the salesman of the caller.

10, 11: The salesman is detached and the caller attached whilst the receptionist informs the caller that he is being transferred.

12: The salesman's media is attached, enabling communication to the caller. The receptionist's call leg is released as it is no longer required.

In the following section we use the provided examples to show how the requirements for an Extended Call Control Web Service can be developed.

IV. EXTENDED CALL CONTROL WEB SERVICE

In this section we show how an ECC specification, as proposed in [3] can be developed.

Using the examples provided in figures 3 and 4, we will show how characteristic sequences can be encapsulated in a single method, including associated notifications.

A. Requirements

Considering Multi-Party Call Control as defined in [9], and considering a Parlay-X Web Service being implemented as shown in figure 1, i.e. as a form of Parlay application, the following basic requirements for an ECC Web Service emerge:

- 1) The Web Application has to be able to maintain the state of the call, and as such a session identifier is required that is unique for each call. This would allow control to be passed from one Web Application to another, and in the case of the Web Application failing, due to an interruption in the communication or other reasons, being able to recover the session.
- 2) A Web Application using Extended Call Control Web Service would have to be able to subscribe to receive notifications. This particular topic is not covered by this paper and requires further work.
- 3) Multi-party calls have to be represented in a manner that whilst maintaining a suitable level of abstraction does not prevent the application from differentiating between the various parties and controlling parties or their connections separately.
- 4) The Parlay-X ECC Web Service must be mapped to various base Parlay SCF methods, which themselves facilitate advanced call control.
- 5) A call model representing the call in an abstracted readily understandable manner.

B. ECC Methods

Considering figure 3. Initially when a Web Application registers with the ECC Web Service a large sequence of diagrams are required to register with the Parlay gateway framework and create triggers within the network to monitor for applicable events. The Web Application should be able to provide the required triggers in the initial *logon* method. The ECC WS would then create the appropriate objects and inform the Application.

The Application should not be required to specifically create any call control objects, only indicate required connections to the ECC WS, as shown in message sequences 8 to 14. Predefined service agreements should be used to determine charging plans. In addition the ECC WS should automatically complete all call processing, as shown in message 15. A single *connect* method should be sufficient to establish a call leg and appropriate routing within the network.

Notifications arising from direct Application requests and relating to ultimate success or failure of the call should be reported, however intermediary notifications are not necessary.

Considering figure 4, control of the state of a participant is required and their connection to other parties. Alteration of the media stream should be extensible depending on the type of media in use (Multiparty Call Control requires participants to use the same media [8, pg. 174]), as shown when the application wishes to *mute* a particular participant in which a Multi-Party Call mapping is required the *detachMediaReq()* would be used, instead of the *MediaStreamAllow()* method when video input is involved.

If a primary party is leaving a call, as shown in message sequences 12–13, when the receptionist leaves the call and is no longer in communication with the parties, the ECC WS should ensure that other parties of the call can communicate. These operations would be grouped into a *disconnect* method.

Thus by means of simple abstracted methods, it is possible to provide advanced call control use cases.

C. ECC Call Model

A call model is required to represent both parties and the state of their connections, as identified in section IV-A, thus a far richer set of service functionality becomes available, than that offered by a simple request response type [7].

This call model must be simplistic but have the potential of offering varying level of abstraction as required by the Application. We propose an adaption of the Services for Computer Supported Telecommunications Applications (CSTA) relationship between calls, devices, and connections as a suitable call model as shown in figure 6.

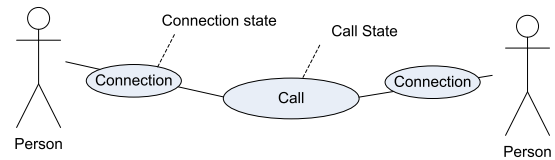


Fig. 6. Abstracted Model of Call for ECC Web Services

A person is defined as the representation of the entity with which communication will ultimately take place. The call represents the relationship between one or more persons. The connection represents the state of the connection between the person and the call.

The state of the MultiParty Call object would be mapped to the Web Application call object and the state of the MultiParty Call leg objects would be mapped to the Web Application connection object.

V. CONCLUSION

We have proposed an extensible Extended Call Control specification based on identification of key characteristic sequences in the base Parlay methods. An example Virtual PBX Web Application is used to show how an ECC Web Service supports advanced Web based call control. A call model suitable for use in the Extended Call Control Web Service is proposed.

REFERENCES

- [1] Ericsson. Meeting #32: Parlay X Call Control improvement, C5-050482. Technical report, Joint Working Group (Parlay, ETSI TISPAN Project OSA, 3GPP CT5), September 2005.
- [2] Appium. Meeting #33: Parlay X 3.0 Enhanced Call Control, C5-050616. Technical report, Joint Working Group (Parlay, ETSI TISPAN Project OSA, 3GPP CT5), October 2005.
- [3] Ericsson BT Appium, ETRI. Meeting #34: Requirements Parlay X Etended Call Control, C5-060016. Technical report, Joint Working Group (Parlay, ETSI TISPAN Project OSA, 3GPP CT5), February 2006.
- [4] Z. Lozinski. Parlay/OSA and the Intelligent Network. February 2005.
- [5] H. Hanrahan. *Network Convergence: Services, Applications, Transport and Operations Support*. Wiley, New Jersey, USA, first edition, 2007.
- [6] B. Fricke and H.E. Hanrahan. The Development of a Structured Approach to Service Provisioning in a Parlay Environment. In *Proceedings of Southern African Telecommunication Networks and Applications Conference (SATNAC): The network @ work*, Western Cape, South Africa, September 2006.
- [7] D. E. Vannucci and H. E. Hanrahan. Analysis of State Models for Telecommunication Services. In *Proceedings of Southern African Telecommunication Networks and Applications Conference (SATNAC): IP Generation leading the charge for convergence*, George, South Africa, September 2005.
- [8] R. Jain, J.-L. Bakker, and F. Anjum. *Programming Converged Networks*. John Wiley & Sons, Washington, USA, first edition, 2005.
- [9] Parlay ETSI. Open Service Access (OSA); Application Programming Interface (API); Part 4: Call Control; Sub-part 3: Multi-Party Call Control SCF (Parlay 5). January 2007.