

SIP 기반의 VoIP 시스템에서의 확장된 자동 콜백 서비스의 구현

조 현 규^{*} · 이 기 수^{**} · 장 춘 서^{***}

요 약

인터넷 전화나 PSTN(Public Switched Telephone Network)에서의 자동 콜백(Automatic Callback) 서비스는 상대방이 통화 중 상태에 있을 때 이 통화가 종료되면 상대방에 대한 재 다이얼링을 자동으로 시도할 수 있는 유용한 서비스이다. 그러나 기본적인 자동 콜백 서비스에서는 긴급 통화를 요하는 경우에도 자동 콜백을 요청한 사용자는 상대방이 통화를 종료할 때까지 계속 기다려야하는 문제점을 가지고 있다. 본 논문에서는 CPL(Call Processing Language)을 확장시키고 사용자 시스템 내에 이 확장된 CPL을 처리할 수 있는 기능과 SIP INVITE 메시지에 의해 생성된 다이얼로그 상태 정보를 처리하는 다이얼로그 이벤트 패키지(Dialog Event Package)를 포함시켜 두 가지 기능을 연동하는 방식으로 기존의 문제점을 해결하였고 SIP(Session Initiation Protocol) 기반의 VoIP(Voice over IP) 시스템에서 사용할 수 있도록 구현하였다.

Implementation of Extended Automatic Callback Service in SIP-based VoIP System

Hyun-Gyu Jo^{*} · Ky-Soo Lee^{**} · Choon-Seo Jang^{***}

ABSTRACT

On the internet phone or PSTN(Public Switched Telephone Network), the automatic callback is an useful service in the case of busy state when one user calls the other. By using this service, automatic redial is possible when the other party hangs up. However, in the basic automatic callback service, the user who wants callback should wait until the other party hangs up even in the case of emergency. Therefore in this paper, to solve this problem we have extended CPL(Call Processing Language) and, within user system, we have included and linked this extended CPL processing module and dialog event package which processes SIP INVITE initiated dialog state informations. We have implemented this system for being used in SIP(Session Initiation Protocol)-based VoIP(Voice over IP) system.

키워드 : SIP, 자동 콜백 서비스(Automatic Callback Service), 다이얼로그 이벤트 패키지(Dialog Event Package), CPL

1. 서 론

VoIP(Voice over IP) 기술을 활용한 인터넷 전화는 데이터와 음성회선을 통합할 수 있고 기존의 PSTN(Public Switched Telephone Network)을 통한 전화 서비스 보다 상대적으로 저렴하게 이용할 수 있는 등 여러 장점들을 가지고 있어 이용이 증가하는 추세이다. 따라서 이에 적용할 수 있는 다양한 부가적인 서비스들에 대한 연구도 활발히 이루어지고 있다. 그 가운데 자동 콜백 부가서비스는 통화를 원하는 상대방이 통화 중일 경우 수동으로 계속 재 다이얼링

을 시도하는 불편함 대신에 상대방의 현재 통화가 종료되면 이를 통보받고 자동으로 호를 요청할 수 있는 유용한 기능이다.

그러나 기본적인 서비스에서는 자동 콜백을 요청한 사용자가 상대방이 언제 통화를 종료할지 알 수 없고 일반적으로 호가 설정된 후부터 종료까지는 오랜 시간이 소요될 수도 있으므로 긴급 통화를 요하는 경우에도 호가 종료되기를 계속 기다려야하는 불편함이 존재한다. 또한 부가서비스 가운데 하나인 발신자 표시 서비스를 이용하여 호 요청을 시도한 사용자를 식별할 수 있더라도 급한 용무의 전화인지 등의 의사를 알 수 없으므로 전적으로 호 요청을 수신한 사용자의 임의적인 판단에 의존하게 된다.

SIP(Session Initiation Protocol)[1]을 기반으로 하는 VoIP 시스템에서 기본적인 자동 콜백 부가 서비스는 다이얼로그

※ 본 연구는 금오공과대학교 학술연구비에 의해 연구된 논문임.

* 준 회원 : 금오공과대학교 대학원 컴퓨터공학과

** 정 회원 : 금오공과대학교 컴퓨터공학부 교수

*** 정 회원 : 금오공과대학교 컴퓨터공학부 교수

논문접수 : 2004년 9월 8일, 심사완료 : 2004년 12월 28일

이벤트 패키지(Dialog Event Package)[2]를 이용하여 SIP INVITE 메소드에 의해 시작된 다이얼로그 상태가 변화될 때마다 바뀐 정보에 대해 통보를 받는 방식으로 구현이 가능하지만 이 기능만으로는 호가 종료될 때까지 계속 기다려야 하는 불편함은 남게 된다.

본 논문에서는 CPL(Call Processing Language)[3] 내의 Signaling Operation에 제안하는 서비스의 호 처리와 관련된 새로운 노드들을 추가하여 CPL을 확장시키고 사용자 시스템에 이 확장된 CPL을 처리할 수 있는 기능과 다이얼로그 이벤트 패키지를 포함시켜 서로 연동하는 방법으로써 기존의 문제점을 해결하였다. 제안된 방법으로 구현된 시스템에서는 호 진행 또는 호 설정 상태에서 자동 콜백 등록자로부터 긴급 통화 시도와 같은 의사를 수용할 수 있고 요청을 받은 사용자 시스템에서 상황에 맞게 적절한 호 제어를 자동으로 수행하는 보다 유연한 서비스를 제공한다.

본 논문의 구성은 다음과 같다. 2장에서 관련 연구로서 SIP의 개요, 다이얼로그 이벤트 패키지, CPL의 특성 및 기존의 자동 콜백 서비스 방식의 문제점을 설명하였다. 3장에서는 기존의 문제점을 해결한 구현된 시스템의 동작 방식 및 전체적인 구조 등을 다루며 4장에서 결론을 맺는다.

2. 관련 연구 및 기존 자동 콜백 서비스의 문제점

2.1 SIP

SIP는 하나 이상의 참가자간에 인터넷 텔레폰 콜, 멀티미디어 컨퍼런스 등의 세션을 생성, 변경, 종료하는 응용 계층의 시그널링 프로토콜이다[4]. 현재 인터넷 전화를 지원하기 위한 대표적인 호 설정 프로토콜로서 H.323과 SIP를 꼽을 수 있으며 특히 SIP는 HTTP와 유사하게 텍스트 형태의 메시지를 사용함으로써 여러 다양한 서비스로의 확장이 용이한 장점을 가지고 있는 프로토콜이다[5].

SIP 기반의 서비스를 제공하기 위한 구성 요소로는 UA(User Agent)와 SIP 서버들이 있다. UA는 UAC(User Agent Client)와 UAS(User Agent Server)로 구분되고 호 설정을 위한 메시지를 송수신하여 SIP 트랜잭션을 시작 및 종료한다. SIP 서버로는 프록시(Proxy) 서버, 리다이렉트(Redirect) 서버, 로케이션(Location) 서버, 레지스트라(Registrar) 등이 있다. 프록시 서버는 UA로부터 받은 메시지를 파싱 및 처리하여 적절한 응답 메시지를 돌려주거나 다른 서버나 UA로 메시지를 포워딩한다. 리다이렉트 서버는 받은 메시지에 대한 해당 주소를 전달하며 로케이션 서버는 사용자의 위치 정보 등을 제공하고 레지스트라는 UA를 등록, 수정, 삭제하는 역할을 담당한다.

SIP 메시지는 크게 UAC에서 UAS로 전송하는 요청(Request) 메시지와 UAS에서 UAC로 보내는 응답(Response) 메시지로 구분된다. 요청 메시지의 종류에는 INVITE, ACK, CANCEL, BYE, REGISTER, OPTIONS 등이 있으며 응답 메시지는 숫자로 된 상태 코드인 1xx, 2xx, 3xx, 4xx, 5xx, 6xx 계열로 구분된다. SIP 메시지의 바디 부분은 통신에 사

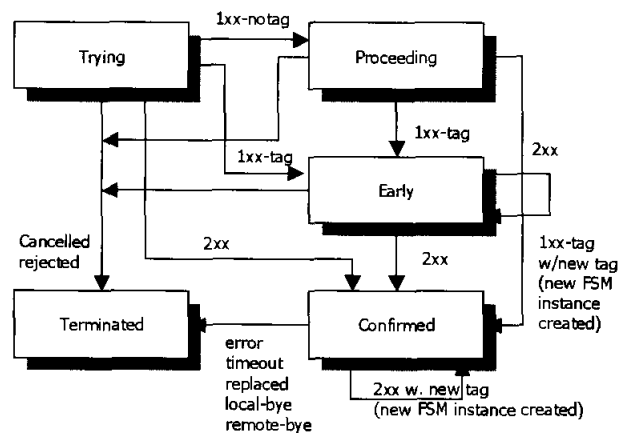
용될 미디어의 협약에 필요한 세션 정보를 기술하는 SDP(Session Description Protocol)[6]로 표현되며 호 설정 후 실제 통신은 RTP(Real-time Transport Protocol)[7] 상에서 이루어진다. RTP는 오디오, 비디오 및 시뮬레이션 데이터와 같은 실시간 데이터 전송하는데 적합한 프로토콜이다.

2.2 다이얼로그 이벤트 패키지

다이얼로그 이벤트 패키지는 다른 사용자의 다이얼로그 상태 정보를 기반으로 등록(Subscription)과 알림(Notification) 기능을 제공하며 자동 콜백 서비스뿐만 아니라 SIP 기반의 온라인 회의나 인스턴트 메시지 서비스에도 유용하게 적용할 수 있다.

SIP에서 사용되는 다른 이벤트 패키지와 마찬가지로 다이얼로그 이벤트 패키지도 RFC 3265[8] 문서 내용을 따르며 등록과 알림에 해당하는 SUBSCRIBE 와 NOTIFY 요청 메시지를 사용한다. 다이얼로그 이벤트 패키지 이름은 dialog를 사용하며 이벤트 등록 시 SUBSCRIBE 요청 메시지의 Event 헤더에 들어가고 이때 파라미터로서 call-id, form-tag, to-tag를 필수적으로 포함한다. SUBSCRIBE 요청 메시지는 바디를 생략하거나 포함할 수 있으며 IETF (Internet Engineering Task Force) 워킹 그룹의 드래프트 문서에서는 바디의 사용 시 알림을 필터링하는 용도로 제안하고 있다.

NOTIFY 메시지 통보는 INVITE가 보내지거나 새로운 다이얼로그가 생성될 때 또는 기존의 다이얼로그의 상태나 특성이 변화될 때 발생한다. 따라서 이벤트 패키지 내에서는 NOTIFY 메시지를 보내는 시점과 이때의 메시지 내용을 정하기 위해 다이얼로그 상태 모델이 필요하며 이는 (그림 1)과 같은 FSM(Finite State Machine) 형태로 나타내어진다.



(그림 1) FSM(Finite State Machine)

(그림 1)에서 FSM의 상태 변화는 UAC가 INVITE를 보내면 Trying 상태가 되고 태그 없는 1xx 메시지를 받으면 Proceeding 상태로 바뀌고 Trying 상태나 Proceeding 상태에서 태그가 있는 180 메시지와 같은 응답을 받게 되면 Early 상태로 변화된다. Early 상태는 상대방 전화벨이 울리

고 있는 상태로 볼 수 있다. UAC가 2xx 메시지를 받으면 Confirmed 상태로 전환되며 이는 현재 통화 연결이 되었음을 의미한다. 만약 Early나 Confirmed 상태에서 다른 태그를 가진 1xx 메시지나 2xx 메시지를 받게 되면 내부적으로 FSM은 새로운 인스턴스를 생성하여 상태를 관리하는데 이러한 경우는 INVITE 포크(fork)와 같은 상황에서 발생한다. 호가 종료되면 FSM의 상태는 Terminated로 변하게 되는데 Trying, Proceeding, Early 상태에서 CANCEL 메시지를 보내어 이벤트 cancelled가 발생하였거나 non-2xx 메시지로 INVITE 가 종료되는 경우 또는 Confirmed 상태에서 BYE 메시지를 송수신하여 통화가 끝남에 따라 다이얼로그가 종료됨을 의미한다.

변화되는 다이얼로그 정보는 application/dialog-info+xml 포맷의 XML(Extensible Markup Language) 문서 형태로 NOTIFY 메시지의 바디에 담아 통보되며 문서의 Namespace는 urn:ietf:params:xml:ns:dialog-info를 사용한다. 다이얼로그 정보의 구조는 최상위 요소인 dialog-info 요소로 시작되고 하위 요소로서 특정 다이얼로그에 대한 정보를 표현하는 dialog 요소를 가지며 이 dialog 요소 내의 state 요소는 다이얼로그의 상태를 나타낸다.

dialog-info 요소는 version, state, entity 속성을 필수적으로 포함한다. 통보되는 문서의 순서를 위해 사용되는 version 속성은 0에서 시작하여 문서가 보내질 때 마다 1씩 증가시킨다. state 속성은 'full' 또는 'partial' 값을 사용하여 문서가 전체 다이얼로그 정보를 포함하는지 또는 이전에 전송한 문서와 비교해 변화된 다이얼로그 정보만 포함하는지를 가리키고 entity 속성은 등록된 사용자의 URI를 나타낸다.

dialog 요소는 현 다이얼로그를 구별하기 위한 id 속성을 필수적으로 포함하며 옵션 속성으로서 call-id, local-tag, remote-tag 및 direction을 가진다. id는 송신자(Caller)의 경우 INVITE 메시지를 보낼 때 생성되고 이후 태그가 있는 1xx 메시지나 2xx 메시지를 수신하여 다이얼로그가 생성되어도 같은 값을 유지한다. 그러나 포크에 의해 추가적인 1xx 메시지나 2xx 메시지가 수신되어 다른 다이얼로그가 생성되면 이 새로운 다이얼로그에는 새로운 id가 할당된다. 수신자(Callee)의 경우 id의 생성은 기존 다이얼로그 이외의 새로운 INVITE 메시지가 수신될 때 발생한다. 그러나 2xx 메시지나 태그가 있는 1xx 메시지를 보내면서 다이얼로그가 생성되어도 이때의 id는 생성된 값에서 변화가 없다.

dialog 요소의 하위 요소인 state 요소는 (그림 1)의 FSM에서 보이는 다이얼로그의 현재 상태를 가지며 옵션 속성인 event로서 상태의 변화를 일으킨 이벤트를 나타낸다.

2.3 CPL(Call Processing Language)

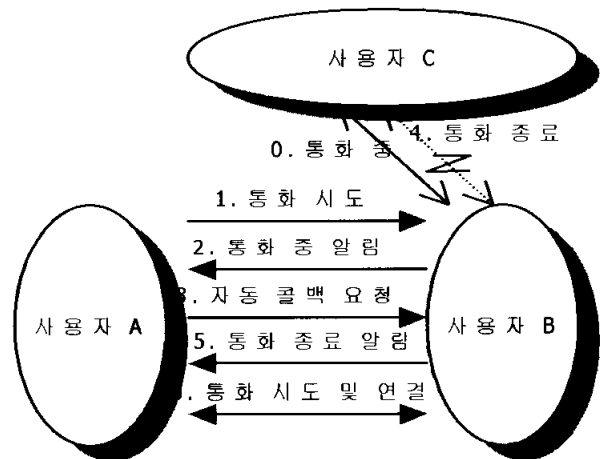
호 처리 언어(CPL)는 인터넷 전화 서비스에서 사용자가 호를 처리하는 다양한 방식을 기술하고 제어할 수 있는 언어이다. CPL은 XML 기반의 언어로서 텍스트 기반 태그로 구성되며 XML 문서들이 갖는 손쉬운 편집기능, 여러 프로토콜을 이용한 전달기능과 확장이 쉬운 장점을 가진다[9].

이러한 CPL의 주요 골격과 요구사항 그리고 CPL의 문법과 관련한 부분들은 RFC 2824[10], RFC 3880 표준 문서로 정의되어 있다.

CPL 스크립트는 호 요청 시 수행할 행동(operation)과 결정(decision)을 명시하는 Call Processing Action과 확장을 위해 예약된 Ancillary Information의 두 가지 정보 타입으로 구성된다. Call Processing Action은 Top-Level Action과 Subaction으로 구분되며 Top-Level Action에서 사용되는 <incoming>과 <outgoing> 태그는 수신된 호에 대한 처리와 호를 송신 시 처리 사항을 명시하고 Subaction은 스크립트의 모듈화와 재사용 부분을 정의하는 용도로서 <subaction> 태그를 사용하여 기술한다. 또한 각 태그의 하위에는 Switches, Location Modifiers, Signaling Operations, Non-Signaling Operations들을 사용하여 조건과 취해야할 행동들에 대한 세부적인 기능을 기술하고 이들의 적절한 조합을 통해 다양한 서비스의 표현 및 처리가 가능하다. 그러나 표준 문서에서 정의하고 있는 기본적인 CPL 기능들의 조합으로도 표현할 수 없는 사용자의 요구를 제공하기 위해서는 CPL의 확장이 필요하다.

2.4 기존의 자동 콜백 서비스 방식 및 문제점

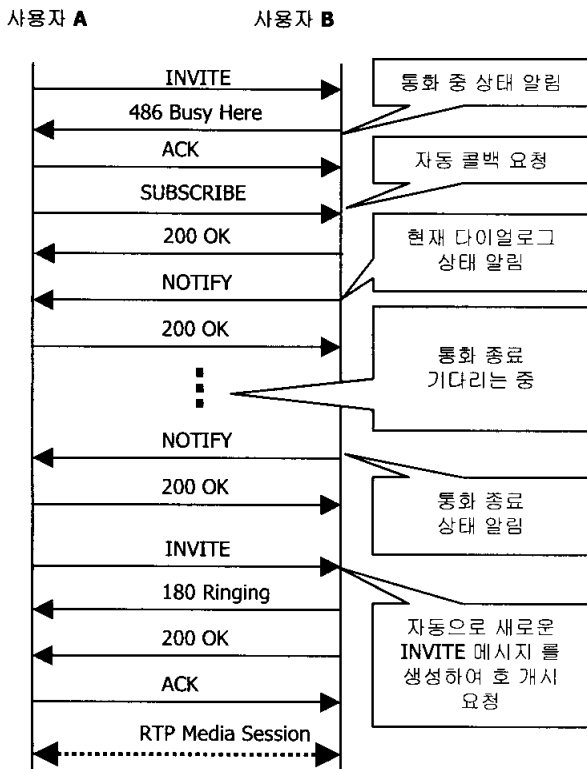
자동 콜백 서비스는 (그림 2)와 같이 사용자 A가 사용자 B에게 통화 요청을 하는 경우 사용자 B가 현재 다른 사용자와 통화 중 상태임을 알게 되면 통화 연결이 될 때까지 수동으로 재 다이얼링하는 불편함 대신에 현재의 통화가 종료되면 이를 통보받고 자동으로 즉시 통화를 시도할 수 있는 유용한 기능이다[11].



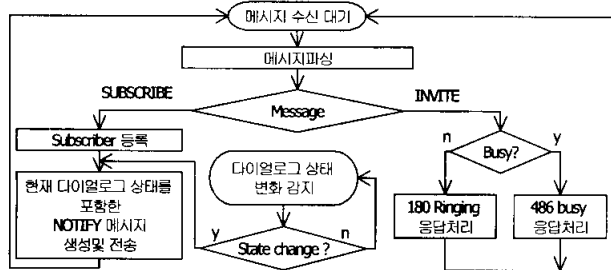
(그림 2) 자동 콜백 서비스

SIP 기반의 VoIP 시스템에서 자동 콜백 서비스의 구현은 변화되는 다이얼로그 정보를 제공하는 기능을 가지고 있는 다이얼로그 이벤트 패키지를 사용하여 가능하며 (그림 3)과 같은 처리 흐름으로 동작한다. 그러나 기본적인 자동 콜백 서비스에서는 (그림 4)의 처리 흐름도와 같이 요청을 받은 사용자 측에서 등록자에게 자신의 현재 및 변화되는 다이얼

로그의 상태만을 전송하여 주기만 한다. 또한 호가 진행 중이거나 호가 성립된 상태에서 받은 호 개시 요청에 대해서는 항상 486 busy 응답 메시지를 보내어 통화중임을 알린다. 따라서 등록자가 새로운 호를 개시하기 위해서는 기존의 호가 종료되는 알림 메시지를 계속 기다려야만 한다. 이는 일반적으로 호가 설정된 후부터 종료까지는 오랜 시간이 소요될 수도 있으므로 긴급 통화를 요하는 경우에도 호가 종료되기를 계속 기다려야만 하는 불편함이 존재함을 의미한다.



(그림 3) 기존의 자동 콜백 서비스 호 처리 흐름도



(그림 4) 자동 콜백 요청에 대한 기존의 메시지 처리

3. 시스템 구현

본 논문에서는 사용자 시스템에 확장된 CPL 처리 기능과 다이얼로그 이벤트 패키지를 포함하여 연동시키는 방법으로 기존의 자동 콜백 서비스의 문제점을 해결하였다. 구현된 시스

템에서는 등록자가 상대방의 다이얼로그 상태가 Terminated로 바뀔 때까지 무작정 기다리지 않고 호 진행 또는 호 설정 상태에서도 긴급 통화와 같은 의사 표명이 가능하며 요청을 받은 사용자 시스템에서 자동으로 상황에 따른 호 처리의 진행이 가능하다. 본문의 아래는 구현된 시스템의 내부 구조, 상호 관계 및 주요 모듈 그리고 제안하는 서비스를 처리하는 방식과 내부 동작을 기술하였다.

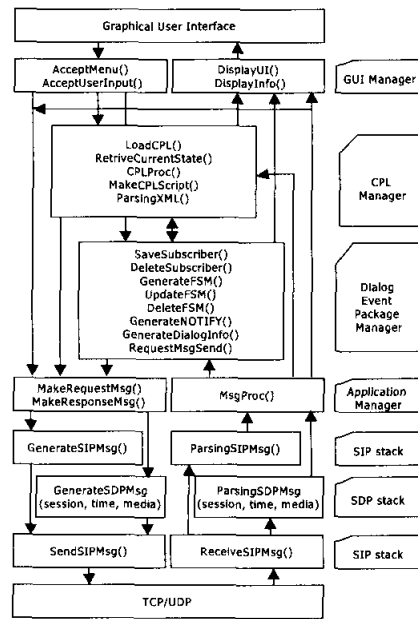
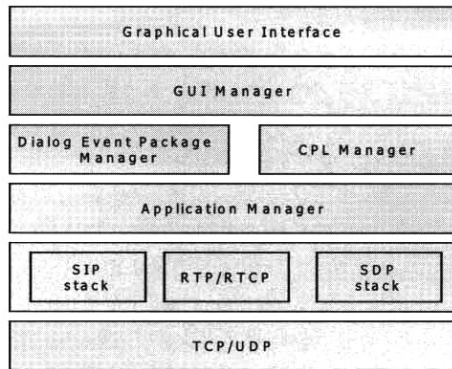
시스템 구성요소인 프록시 서버와 레지스트라는 리눅스에서 운영하였다. UA를 포함한 사용자 시스템은 자바를 이용하여 구현하였고 그래픽 사용자 인터페이스를 사용하여 사용자의 편의성을 높였다. 시스템내의 필요한 모든 인증 처리는 HTTP 다이제스트 인증[12]을 사용하였다.

3.1 사용자 시스템 구조

본 논문에서 구현한 사용자 시스템은 UA, GUI 매니저, 어플리케이션 매니저 및 제안하는 서비스 처리를 위한 다이얼로그 이벤트 패키지 매니저와 CPL 매니저로 구성하였다. UA는 SIP 메시지를 생성하여 전송하거나 수신된 메시지를 파싱 및 처리하며 트랜잭션을 관리한다. 다이얼로그 이벤트 패키지 매니저는 호의 개시부터 종료까지 변화되는 다이얼로그 상태를 관리하고 알림이 발생하면 자동 콜백 서비스를 요청한 사용자에게 다이얼로그 상태 정보를 포함한 NOTIFY 메시지를 생성하여 UA가 알림 메시지를 전송하도록 지시한다. CPL 매니저는 사용자가 GUI를 이용하여 작성한 CPL 스크립트를 생성시키고 CPL 처리가 필요한 경우 레지스트라로부터 스크립트를 가져와 조건에 맞는 해당 호 처리를 지시한다. 통상적으로 CPL의 처리는 프록시 서버에서 CPL 서버에 등록된 스크립트를 해석하여 적절한 호 처리를 진행하는데 본 논문에서는 사용자 시스템의 CPL 매니저가 스크립트를 가져와 해석하여 UAS가 취할 행동들을 지시하도록 기능을 추가하였다. CPL 스크립트는 사용자가 레지스트라에 등록 시 REGISTER 요청 메시지의 바디에 포함시켜 함께 등록하도록 하였다. GUI 매니저는 사용자가 그래픽 사용자 인터페이스를 통해 입력한 사항에 대해 어플리케이션 매니저에게 기능 수행을 지시하거나 어플리케이션 매니저로부터 받은 처리 과정 및 결과 등을 화면에 보여준다. 어플리케이션 매니저는 UA의 기능 모듈을 호출하거나 UA가 수신한 메시지의 형태에 따라 상위로 해당 처리를 요청한다. (그림 5)는 이러한 사용자 시스템의 구조와 내부 동작을 나타내며 <표 1>에는 각 구성 요소가 처리하는 주요 모듈의 용도를 보였다.

3.2 자동 콜백 등록 및 알림

다이얼로그 이벤트 패키지 매니저가 변화되는 다이얼로그 정보와 다이얼로그 이벤트 패키지에 등록하는 등록자를 관리하기 위해 구성된 각 테이블의 구조와 필드의 구성은 <표 2>와 같다. 테이블 FSMT은 호가 개시되는 시점부터 다이얼로그 정보와 변화되는 다이얼로그 상태를 저장, 갱신 및 삭제하는 용도이다. 테이블 DEST는 자동 콜백을 요청한 등



(그림 5) 사용자 시스템 구조 및 내부 동작

<표 1> 사용자 시스템 구성 요소내의 주요 모듈

요소	모듈	용도
GUI Manager	AcceptMenu()	사용자의 메뉴클릭에 따라 Application Manager에게 처리 지시
	AcceptUserInput()	사용자의 입력에 따라 Application Manager에게 처리 지시
	DisplayInfo() DisplayUI()	호의 진행상황이나 받은 메시지를 화면에 출력 메뉴관련 다이얼로그 생성 및 화면에 출력
Dialog Event Package Manager	SaveSubscriber()	다이얼로그 이벤트 패키지에서 subscriber의 등록
	DeleteSubscriber()	subscriber의 삭제
	GenerateFSM()	FSM instance 생성
	DeleteFSM()	다이얼로그 상태의 갱신
	GenerateNOTIFY()	다이얼로그의 종료 시 FSM instance 삭제
	GenerateDialogInfo() RequestMsgSend()	알림 발생 시 NOTIFY 메시지 생성을 위한 파라미터 생성 다이얼로그 상태 정보의 생성 NOTIFY 메시지 전송 요청
CPL Manager	LoadCPL()	레지스트라에게 CPL 점의 및 스크립트 로딩
	RetriveCurrentState()	다이얼로그이벤트패키지에서 현재 상태를 확인
	MakeCPLScript()	CPL 스크립트 처리
	ParsingXML()	CPL 스크립트 생성 XML 파싱
Applicati on Manager	MakeRequestMsg()	사용자의 요구 또는 다이얼로그 이벤트 manager의 지시에 따라 UA에게 요청 메시지 생성 지시
	MskResponseMsg() MsgProc()	UA에게 필요한 응답 메시지 생성 지시 UA가 받은 메시지에 따라 상위로 처리를 요청
SIP stack	GenerateSIPMsg()	SIP 요청 메시지 또는 응답 메시지 생성
	ParsingSIPMsg()	받은 SIP 메시지 파싱
	SendSIPMsg()	SIP 메시지 전송
	ReceiveSIPMsg()	SIP 메시지 수신
SDP stack	GenerateSDPMsg()	SDP 생성
	ParsingSDPMsg()	SDP 파싱

<표 2> 다이얼로그 이벤트 패키지를 위한 테이블 구성

테이블 이름	테이블의 용도	필드 구성
FSMT (Dialog Finite State Machine Table)	변화되는 다이얼로그 상태를 저장 및 관리	id(String), state(String), call_id(String), local_tag(String), remote_tag(String), direction(String), event(String), expire(int)
DEST (Dialog Event Subscription Table)	이벤트 패키지에 등록된 등록자(subscriber)의 저장, 수정 및 삭제	dsbsubscriber(String), dsbsubscribertag(String), dnotifier(String), dnotifiertag(String), dcontact(String), dcallid(String), dseq(int), dversion(int), dexpire(int), dstate(String)

```

C:\Program Files\WEditPlus\ZWlauncher.exe
NOTIFY sip:008@ip2.kumoh.ac.kr SIP/2.0
Via: SIP/2.0/UDP 202.31.137.177;branch=29HG4bk1392337410
Max-Forwards: 69
From: sip:johg@ip2.kumoh.ac.kr;tag=1364790975
To: sip:008@ip2.kumoh.ac.kr;tag=1822781707
Call-ID: 334376915
Event: dialog
Subscription-State:active;expires=3600
Content-type: application/dialog-info+xml
Seq: 1 NOTIFY
Content-Length: 409

<?xml version="1.0"?>
<dialog-info xmlns="urn:ietf:params:xml:ns:dialog-info"
  version="0"
  state="full"
  entity="sip:johg@ip2.kumoh.ac.kr">
  <dialog id="346220475" call-id="786081557"
    local-tag="1590028460" remote-tag="325926277"
    direction="initiator">
    <state>confirmed</state>
  </dialog>
</dialog-info>
    
```

(그림 6) 다이얼로그 정보를 포함한 NOTIFY 메시지의 예

등록자의 정보와 서비스 유효시간 및 알림에 해당하는 NOTIFY 메시지의 버전 값 등을 관리한다.

다이얼로그 이벤트 패키지 매니저는 SUBSCRIBE 메시지

를 수신하면 인증을 거쳐 테이블 DEST에 자동 콜백 요청자를 등록시킨다. 다음 현재 다이얼로그 상태를 테이블 FSMT에서 가져와 application/dialog-info+xml 포맷으로 생성하고 NOTIFY 메시지의 바디에 담아 통보한다. 이후 서비스 유효 시간 내에서 다이얼로그 상태나 특성이 변화될 때마다 새로운 상태 정보를 담아 해당 등록자에게 재 통보한다. (그림 6)은 application/dialog-info+xml 포맷의 XML 문서를 바디에 포함한 NOTIFY 메시지의 실제 예이다. 여기서 현재 호가 설정된 상태이며 통화 중임을 알 수 있다.

3.3 제안된 새로운 자동 콜백 서비스 구현

본 논문에서는 기존 서비스의 문제점을 해결하기 위해 CPL 내의 Signaling Operation에 제안하는 서비스의 호 처리와 관련된 bye-cancel, call, msg와 같은 새로운 노드들을 추가하여 CPL을 확장시켰고 사용자 시스템에 이 확장된 CPL을 처리할 수 있는 기능을 포함시켜 다이얼로그 이벤트 패키지와의 연동하는 방법을 사용하였다. 즉, 내부적으로 CPL 매니저가 확장된 CPL 스크립트를 처리하여 호 제어를 진행함에 있어 다이얼로그 이벤트 패키지로부터 현재의 다이얼로그 상태 정보를 제공 받고 상황에 맞는 메시지 생성을 UA에게 지시하도록 하였다. 또한 자동 콜백 요청을 받은 사용자 측에서 호를 개시하는 경우에는 CPL 매니저가 다이얼로그 이벤트 패키지에서 생성하는 알림을 생략하는 지시가 가능하도록 하였다.

따라서 기존의 자동 콜백 서비스와는 달리 호의 진행이나 설정 상태인 동안에 긴급 통화 시도와 같은 요청을 받으면 등록자의 요구와 사용자가 레지스트라에 등록해둔 확장된 CPL 스크립트의 내용 및 현재의 다이얼로그의 상태에 맞추어 요청을 받은 사용자 시스템이 호 제어를 자동으로 진행 시킴으로서 기존의 자동 콜백 서비스에서의 불편함을 해결하였다.

자동 콜백 등록자의 의사 표명 방법은 INVITE 요청 메시지 내에 emergency, urgent, normal, non-urgent와 같은 4가지의 우선순위를 명시한 Priority 헤더 필드를 포함시키

면 된다. CPL을 확장시킨 이유는 현재 CPL 표준에서는 제안하는 서비스에 관련된 호 처리 동작에 대해서는 정의되어 있지 않기 때문이다. 새로 추가한 노드인 bye-cancel은 현재 진행 중인 다이얼로그를 종료시키기 위한 노드이며 call은 INVITE 메시지를 생성하여 호 개시를 요청하는 노드이고 msg는 인스턴트 메시지를 생성하여 통화를 종료시킨 상황 등을 상대방에게 통보하는 노드이다.

(그림 7)은 확장된 CPL 처리 기능과 다이얼로그 이벤트 패키지 기능의 연동을 위해 구성된 구조 및 동작이고 (그림 8)은 자동 콜백 등록자로부터 Priority 헤더 필드를 포함한 INVITE 요청 메시지를 받은 사용자 시스템에서의 호 처리를 진행하는 흐름도이다.

(그림 8)의 확장된 CPL 처리에서 bye-cancel은 호 종료 처리를 위한 노드로서 다이얼로그 상태가 Confirmed이면 BYE 메시지를 생성하고 호의 진행 상태 중일 때는 CANCEL 메시지를 생성하여야 하므로 CPL 매니저는 다이얼로그 이벤트 패키지로부터 현재의 다이얼로그 상태를 제공받아 상황에 맞는 메시지 생성 및 해당 처리를 UAS에게 지시한다. call은 긴급 통화 요청을 한 등록자에게 호의 종료를 알리는 대신 요청을 받은 사용자 측에서 호 개시를 하는 노드로서 이때는 다이얼로그 이벤트 패키지에서 알림 메시지를 전송하지 않도록 지시한다.

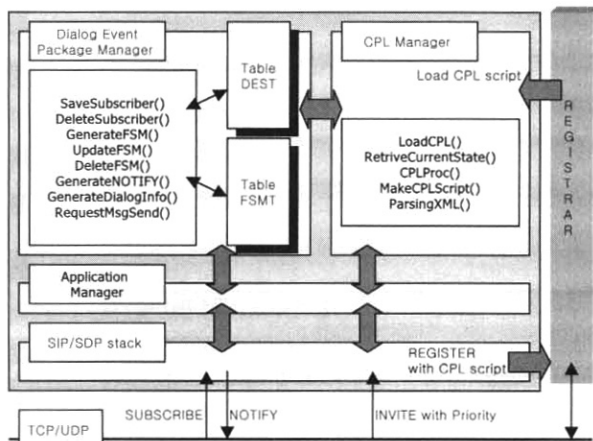
<표 3>은 CPL에 추가한 노드들의 구문이다. 사용자는 이 추가된 새로운 노드들과 CPL의 기본 기능을 조합하여 UAS가 취해야할 행동들을 확장된 CPL 스크립트로 작성하여 레지스트라에 등록할 수 있도록 하였다.

<표 3> 추가된 Signaling Operation 노드들의 Syntax

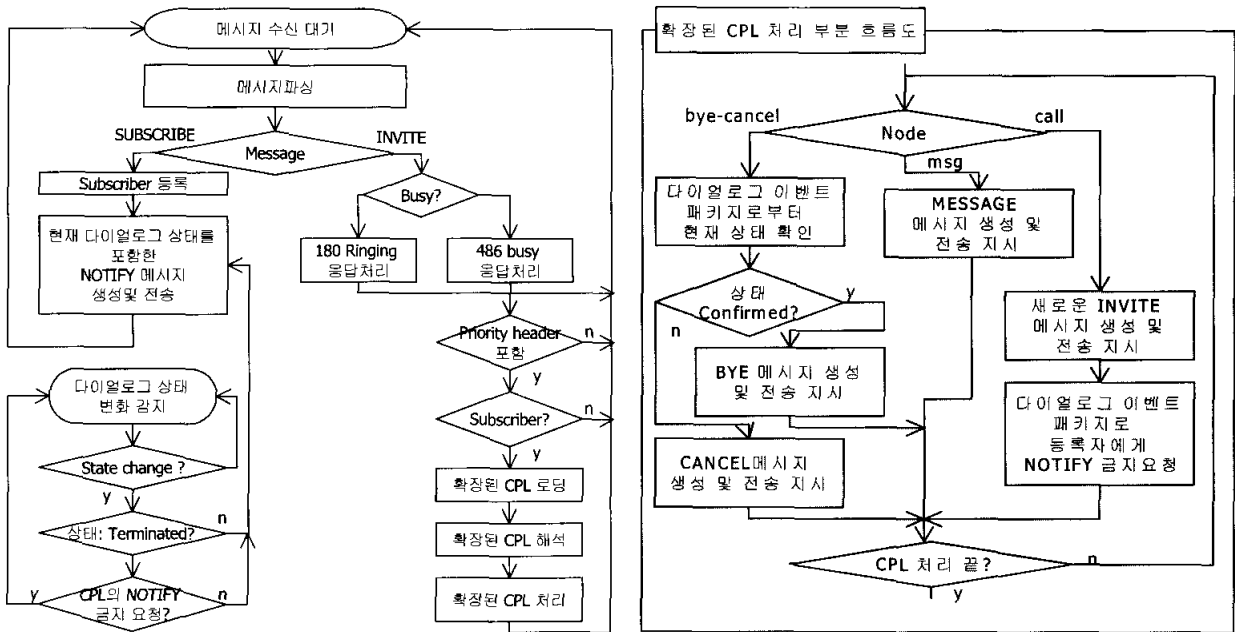
Node : bye-cancel	Node : call	Node : msg
Output : None	Output : None	Output : None
Next Node: None	Next Node: None	Next Node: None
Parameters : None	Parameters: None	Parameters: "contents" optionally allows the script to specify a reason for the MESSAGE sending

확장된 CPL 스크립트는 우선순위에 따라 모든 사용자들에게 동일한 처리를 적용시킬 수도 있지만 사용자들을 구분하여 특정한 사용자의 요청에 대해 특정한 처리를 명시할 수도 있다. 예를 들어 회사의 상관으로부터 받은 긴급 통화 요청은 즉각 대응이 가능하도록 하거나 하위 직원으로부터의 요청으로 확인되면 현재의 호가 종료될 때까지 보류시킬 수도 있음을 의미한다.

(그림 9)는 사용자와 우선순위에 따라 호 처리를 달리하도록 작성한 확장된 CPL 스크립트의 예이다. (그림 9)에서 ①은 urgent 이상의 우선순위를 포함한 호 요청을 받을 경우 CPL 매니저가 UAS에게 현재 진행 중인 호를 종료시키고 호를 종료한 이유를 기술한 인스턴트 메시지를 생성하여 통화 중이었던 사용자에게 전송하고 긴급 통화를 요청한 사용자에게 INVITE 요청 메시지를 직접 보내도록 지시하는 스크립트 부분이다. ③은 가장 높은 우선순위인 emergency



(그림 7) 제안된 서비스를 위한 내부 구조 및 동작



(그림 8) 개선한 자동 콜백 서비스의 처리 흐름도

분이 가능해진다.

```

<?xml version="1.0" encoding="UTF-8"?
<cpl xmlns="urn:ietf:params:xml:ns:cpl"
  xmlns:irt="http://www.irt.kumoh.ac.kr/dialog-request"
  xmlns:xsi="http://www.w3.org/2002/XMLSchema-instance"
  xsi:schemaLocation="urn:ietf:params:xml:ns:cpl:cpl.xsd
  http://www.irt.kumoh.ac.kr/dialog-request:dialog-request.xsd"
  <subaction id="boss">
    <priority-switch>
      <priority greater="urgent">
        <irt:bye-cancel/>
        <irt:msg contents="I received an urgent call"/>
        <irt:call/>
      </priority>
      <otherwise>
        <irt:msg contents="Please wait"/>
      </otherwise>
    </priority-switch>
  </subaction>
  <subaction id="friend">
    <priority-switch>
      <priority greater="urgent">
        <irt:bye-cancel/>
        <irt:msg contents="I received an urgent call"/>
        <irt:call/>
      </priority>
      <otherwise>
        <irt:msg contents="Please wait"/>
      </otherwise>
    </priority-switch>
  </subaction>
  <subaction id="anonymous">
    <priority-switch>
      <priority greater="urgent">
        <irt:msg contents="Please wait"/>
      </priority>
    </priority-switch>
  </subaction>
  <incoming>
    <address-switch field="origin">
      <address is="sip:csjang@kumoh.ac.kr">
        <sub ref="boss"/>
      </address>
      <address is="sip:kmh@kumoh.ac.kr">
        <sub ref="friend"/>
      </address>
      <otherwise>
        <sub ref="anonymous"/>
      </otherwise>
    </address-switch>
  </incoming>
</cpl>
  
```

(그림 9) 확장된 CPL 스크립트 예

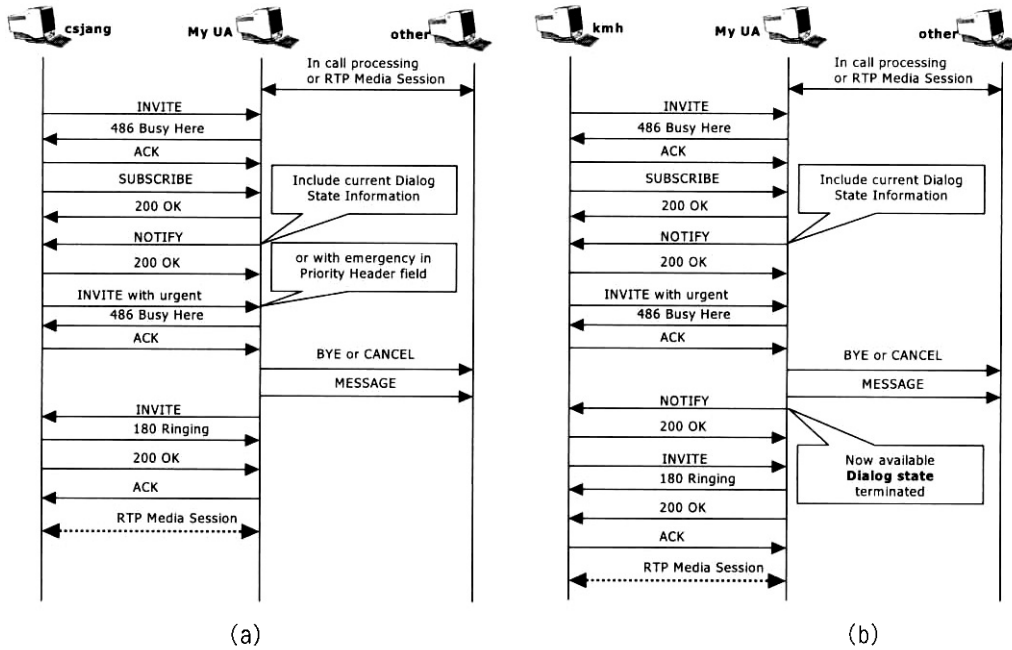
를 포함한 요청을 받을지라도 인스턴트 메시지의 전송만 수행하고 자동으로 호를 종료시키지 않으며 호의 종료를 사용자가 직접 판단할 수 있도록 한다. 이와 같은 처리는 ④와 같이 address-switch 태그 내에 즉각 대응할 사용자와 그렇지 않은 사용자를 명시함으로써 요청한 사용자들의 구

3.4 제한된 서비스의 동작 예

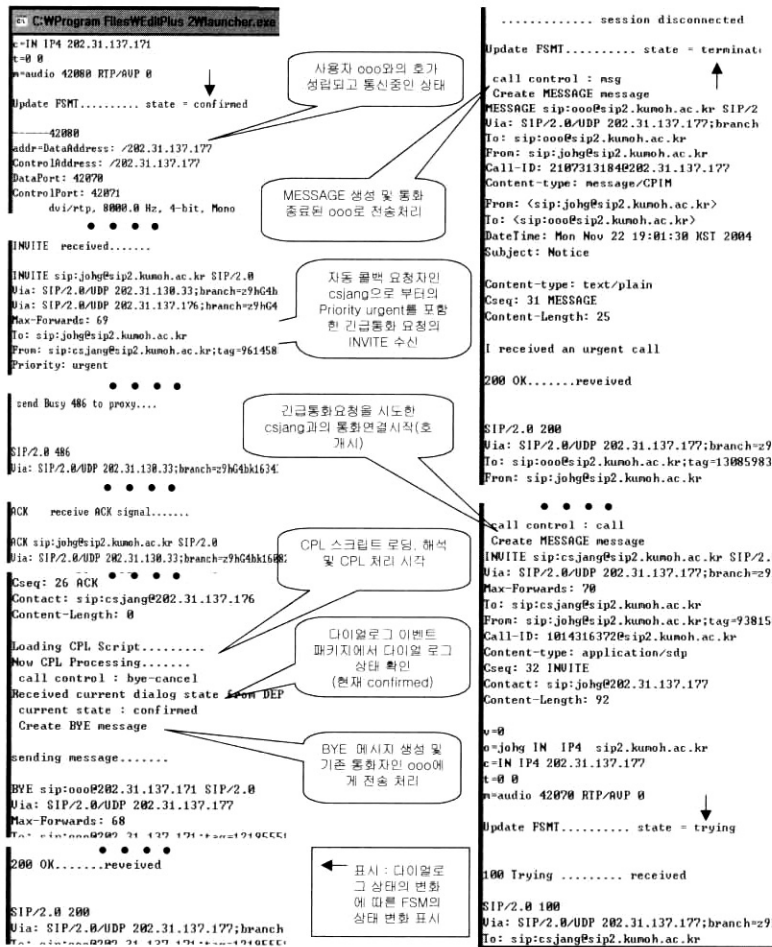
(그림 10)은 (그림 9)와 같은 CPL 스크립트를 해석하여 사용자를 구분하고 호 처리를 진행하는 흐름도의 예이다. (그림 10)의 (a)는 사용자 csjang으로부터 urgent 이상의 우선순위를 포함한 요청을 받을 경우 현재 다이얼로그의 상태와 상관없이 호를 종료시키며 사용자 csjang에게 호 개시를 직접 요청하고 설정을 이루는 처리 흐름도이다. 여기서 호가 종료됨에 따라 다이얼로그 상태가 Terminated로 바뀌지만 CPL의 call 동작 처리에 따라 다이얼로그 이벤트 패키지에서는 알림 발생 과정이 생략되며 INVITE 요청을 받은 등록자(csjang)측에서는 이를 승락하여 양자간에 즉시 통화가 이루어진다. (그림 10)의 (b)는 사용자 kmh로부터의 요청에 대해 (그림 9)의 ② 같이 call 노드를 명시하지 않은 처리가 진행되는 흐름도이다. 이와 같이 call 동작을 명시하지 않은 경우에는 다이얼로그 이벤트 패키지에서 알림이 발생하여 다이얼로그가 종료되었음을 통보하고 등록자 측에서 INVITE 메시지를 자동으로 생성하여 호 개시를 요청한다. (그림 10)의 (a) 경우는 현재 통화의 종료를 알리는 단계가 생략됨으로 (그림 10)의 (b) 보다 빠른 호 처리가 가능하지만 호 개시를 자동 콜백 요청을 받은 측에서 시작함으로써 전화 사용료를 부담하게 될 수도 있다. (그림 11)은 구현된 시스템에서 긴급 통화 요청을 받아 이를 처리하는 실제 동작 과정을 캡처한 화면이다.

3.5 사용자 인터페이스

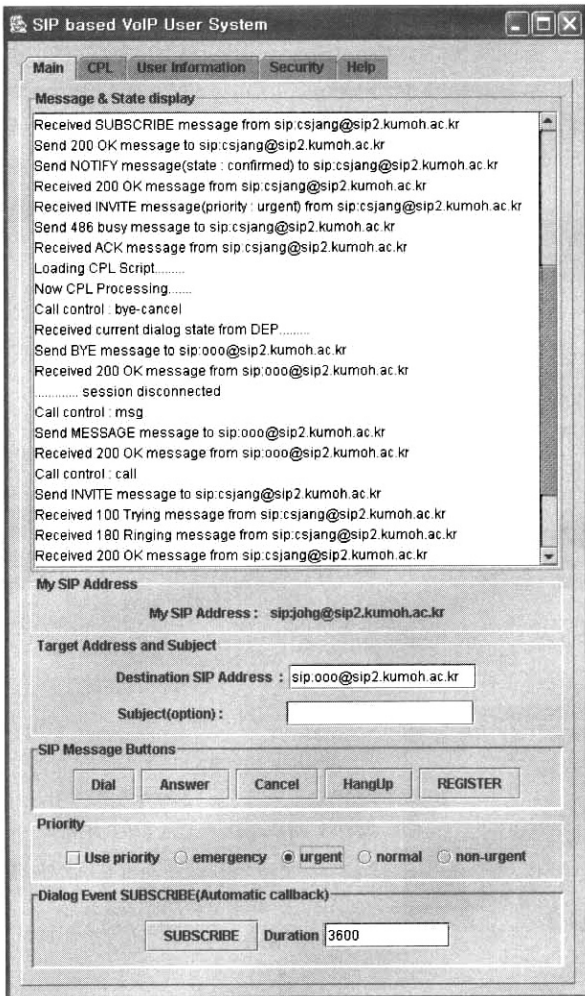
(그림 12)는 시스템 내에서 구현한 사용자 인터페이스들이다. (그림 12)의 (a)는 사용자 시스템의 메인화면으로서



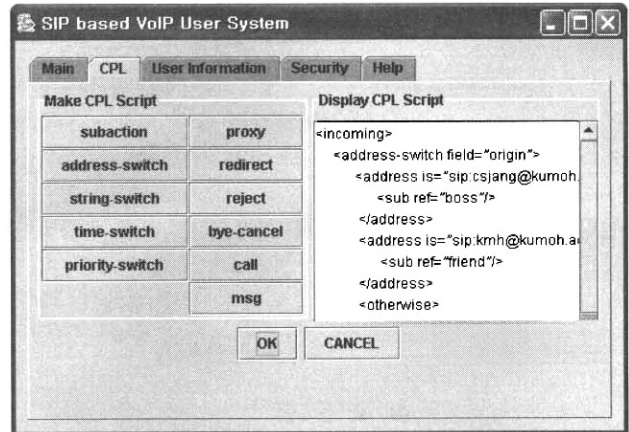
(그림 10) 제안된 서비스의 호 처리 흐름 예



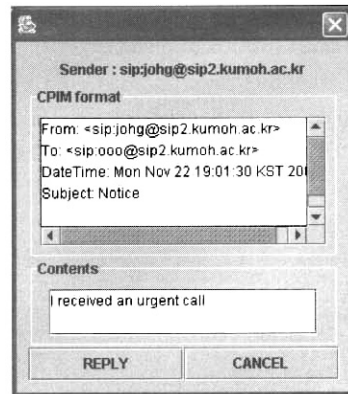
(그림 11) 제안된 서비스로 호 처리 진행의 실제 화면



(a) 사용자 시스템의 메인화면



(b) CPL 스크립트 작성을 위한 인터페이스



(c) 인스턴트 메시지 출력화면

(그림 12) 사용자 인터페이스

레지스트라에 등록하거나 전화를 송수신하는데 필요한 입력과 버튼 그리고 자동 콜백 서비스의 요청과 우선순위를 명시하기 위한 선택 버튼 및 사용자가 호의 진행 상태와 과정을 알 수 있도록 표시하는 부분으로 구성하였다. (그림 12)의 (b)는 CPL 스크립트를 작성하기 위한 화면으로서 필요한 태그들을 버튼을 이용하여 생성시키고 세부적으로 필요한 사항들은 사용자가 직접 입력하도록 하였다. (그림 12)의 (c)는 CPL 매니저가 msg 노드를 처리하여 생성한 인스턴트 메시지를 받은 사용자에게 이를 보여주는 출력 화면이다.

4. 결 론

인터넷 전화에서 자동 콜백 서비스는 유용한 부가서비스의 하나이다. 그러나 일반적으로 호가 설정된 후부터 종료까지는 오랜 시간이 소요될 수도 있으므로 이 경우 자동 콜백을 요청한 등록자가 긴급을 요하는 통화를 원할지라도 상대방의 호가 종료될 때까지 계속 기다려야 하는 불편함이 존재한다.

본 논문에서는 SIP기반의 VoIP 시스템에서의 기본적인 자동 콜백 서비스를 SIP INVITE 메소드에 의해 생성된 다이얼로그 상태 정보를 처리할 수 있는 다이얼로그 이벤트 패키지를 이용하여 구현하였고 더불어 기존의 문제점을 해결하기 위하여 CPL 내의 Signaling Operation에 새로운 노드들을 추가하여 CPL을 확장시키고 이 확장된 CPL을 처리할 수 있는 기능을 사용자 시스템 내에 포함시켜 다이얼로그 이벤트 패키지와 연동하는 방법을 제안하였다. 따라서 구현된 사용자 시스템에서는 호 진행 또는 호 설정 상태에서도 자동 콜백 등록자로부터의 SIP INVITE 메시지 내에 우선순위가 명시된 Priority 헤더 필드를 포함한 요청을 수용할 수 있고 CPL 매니저가 등록자의 요구와 사용자가 등록해둔 확장된 CPL 스크립트 내용 및 현재의 다이얼로그의 상태에 따라 상황에 맞게 호 처리를 자동으로 진행할 수 있게 되었다.

내부적으로 CPL의 확장은 Signaling Operation 내에 bye-cancel, call, msg와 같은 새로운 노드들을 추가시켜 사용자들이 새로 추가된 노드들과 기존의 CPL 기능들을 함께

조합하여 제안한 서비스에 필요한 확장된 CPL 스크립트를 생성할 수 있도록 하였다. 사용자 시스템내의 CPL 매니저는 다이얼로그 이벤트 패키지로부터 현재의 다이얼로그 상태를 제공받아 UA에게 필요한 메시지 생성을 지시할 수 있도록 하였고 필요시 알림 메시지 생성 및 전송을 생략할 수 있도록 처리하였다.

인터넷 전화의 수요가 증가함에 따라 이를 위한 다양한 부가서비스의 개발이 요구된다. 향후과제로 유용한 부가 서비스들을 개발함에 있어 CPL을 확장 및 활용할 수 있는 여러 방안을 연구할 예정이다.

참 고 문 헌

[1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, "Session Initiation Protocol," RFC 3261, June, 2002.

[2] J. Rosenberg, H. Schulzrinne, R. Mahy, Ed., "An INVITE Initiated Dialog Event Package for the Session Initiation Protocol (SIP)," draft-ietf-sipping-dialog-package-05, Nov., 4, 2004.

[3] J. Lennox, Xiaotao Wu, H.Schulzrinne, "CPL: A Language for User Control of Internet Telephony Service," RFC 3880, October, 2004.

[4] IETF SIP Working Group, <http://www.ietf.org/html.charters/sip-charter.html>.

[5] Fredrik Fingal, Patrik Gustavsson, "A SIP of IP-telephony," <http://www.cs.columbia.edu/~hgs/sip/papers.html>.

[6] M. Handley, V. Jacobson, "Session Description Protocol," RFC 2327, April, 1998.

[7] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," RFC 1889, January, 1996.

[8] A. B. Roach, "Session Initiation Protocol (SIP)-Specific Event Notification," RFC 3265, June, 2002.

[9] 정옥조, 이일진, 강신각, "VoIP를 위한 CPL 클라이언트 설계 및 구현," 정보처리학회논문지C, 제10-C권 제4호, pp.0501-0508, 2003.

[10] J. Lennox, H Schulzrinne, "Call Processing Language Framework and Requirements," RFC 2824, May, 2000.

[11] A. Johnston, R. Sparks, C. Cunningham, S. Donovan, K. Summers, "Session Initiation Protocol Service Examples," draft-ietf-sipping-service-examples-06, July 16, 2004.

[12] J. Franks, P. Hallam-Baker, J. Hostetler, S. Lawrence, P. Leach, A. Luotonen, L. Stewart, "HTTP Authentication: Basic and Digest Access Authentication," RFC 2617, June, 1999.

조 현 규



e-mail : blackjo@kumoh.ac.kr
 1991년 금오공과대학교 전자공학과(학사)
 1995년 금오공과대학교 대학원 전자공학과(공학석사)
 2002년~현재 금오공과대학교 대학원 컴퓨터공학과(박사과정)

관심분야: SIP, VoIP, 실시간 인터넷 통신

이 기 수



e-mail : kslee@knut.kumoh.ac.kr
 1979년 경북대학교 전자공학과(학사)
 1982년 서울대학교 대학원(공학석사)
 1982년~현재 금오공과대학교 컴퓨터공학부 교수

관심분야: 교육, IT

장 춘 서



e-mail : csjang@kumoh.ac.kr
 1978년 서울대학교 전자공학과(학사)
 1981년 한국과학기술원(공학석사)
 1993년 한국과학기술원(공학박사)
 1981년~현재 금오공과대학교 컴퓨터공학부 교수

관심분야: 실시간 인터넷 통신, VoIP, SIP, 임베디드 시스템