

푸쉬형 음성 서비스를 위한 VoiceXML 플랫폼의 확장

An Extension of the VoiceXML Platform for Push-based Voice Applications

김 경 란*, 홍 기 형**
(Kyoung-Ran Kim*, Ki-Hyung Hong**)

* (주)미디어포드, ** 성신여자대학교 컴퓨터정보학부
(접수일자: 2001년 10월 25일; 채택일자: 2001년 12월 18일)

VoiceXML (Voice eXtensible Markup Language)은 음성 대화를 웹의 마크업 언어로 기술할 수 있는 차세대 대화형 음성 서비스를 위한 표준이다. 현재 발표된 VoiceXML 1.0 사양은 음성대화의 기술에 초점을 두고 있으며, 음성 서비스의 시작 시점에 관해서는 언급이 없다. 본 논문에서는 사용자에 의해 시작하는 주문형 (on-demand) 음성 서비스 (inbound call) 뿐만 아니라, 최근에 고객관계관리 (CRM: Customer Relationship Management) 등에서 요구되는 시스템에 의한 푸쉬형 음성 서비스 (outbound call)가 가능한 VoiceXML 플랫폼을 설계하고 구현하였다. 또한, 본 논문에서는 W3C (World Wide Web Consortium)의 호 제어 요구사항 (Call Control Requirements)을 바탕으로 음성 대화 중에 팩스 송수신이 가능하도록 VoiceXML 문서형 정의 (DTD: Document Type Definition)를 확장하였다.

핵심용어: VoiceXML, 음성 미들웨어, 푸쉬 (Push)형 음성 서비스

투고분야: 음성처리 분야 (2.7)

VoiceXML is a standard dialog mark-up language for the next generation voice applications. The current VoiceXML 1.0 specification is silent on who place outbound calls for push-based voice applications. The push-based voice applications become very important in modern information systems such as CRM. In this paper, we design and implement an extended VoiceXML platform that supports both inbound and outbound voice information services. We also extend the VoiceXML DTD so as to be able to inbound/outbound fax based on Call Control Requirements of W3C.

Keywords: VoiceXML, Voice middle-ware, Push based voice service

ASK subject classification: Speech signal processing (2.7)

I. 서론

VoiceXML은 음성 인식 및 합성 기술과 컴퓨터 전화통합 (CTI: Computer Telephony Integration) 기술에 웹 정보 처리 기술을 접목한 음성 대화 시나리오 저작 표준 언어로 기존 자동응답서비스 (ARS: Automatic Response Service), 상호음성응답 (IVR: Interactive Voice Response) 등의 대화형 음성 서비스 시스템을 대체할 차세대 음성 서비스의 핵심 기술로 인정받고 있다[1-3]. 대화형 음성 서비스

의 경우에 사용자의 정보 접근 필요성에 의하여 사용자가 ARS 등의 음성 서비스 서버에 전화를 걸어서 이루어지는 인바운드 (inbound) 서비스와 정보시스템의 필요에 의하여 사용자에게 전화를 걸어서 이루어지는 아웃 바운드 (outbound) 서비스가 있다.

최근 각광받고 있는 고객관계관리에서는 사용자에게 유용한 정보가 발생하면, 이를 바로 고객에게 전달할 수 있는 보다 적극적인 Push형 대 고객 정보 서비스가 매우 중요하다. 이를 위해서는 전자우편이나 단순 문자 서비스 (SMS: Short Message Service)와 같은 기존 문자 기반 정보서비스가 활용되고 있다. 그러나, 전자우편의 경우, 완전한 Push형 서비스라고 할 수 없다. 그 이유는 전자우

책임저자: 김경란 (krkim@cs.sungshin.ac.kr)
135-988 서울시 강남구 개포동 14-5 도시개발공사빌딩 5층
(주)미디어포드
(전화: 02-3412-2492; 팩스: 02-3412-2491)

편의 확인은 또 다시 사용자의 의지에 달려 있기 때문이다. 단순 문자 서비스의 경우는 휴대전화를 이용하므로 대체로 전달할 수 있는 메시지의 크기가 매우 제한된다. 따라서, 사용자에게 직접 전화를 통하여 접속하여 음성으로 전달할 수 있는 아웃 바운드 음성 서비스가 반드시 필요하다.

그러나 현재 발표된 VoiceXML 1.0[1]에는 아웃 바운드 서비스를 위한 정의가 나타나 있지 않다. 현재 VoiceXML 1.0에서 지원하는 호 제어 (call control)는 호 전환 (call transfer)과 호 해제 (disconnect) 뿐이며, W3C에서 실제 음성 응용에서 필요로 하는 호 제어를 수용하기 위하여 호 제어와 관련한 요구사항을 수집 중에 있다. 또한, 팩스는 현재 대중화된 정보 통신 수단 중의 하나로 음성 정보 서비스와 더불어 사용할 경우, 효율 가치가 매우 높다. 기존 발표된 논문은 음성 인식 및 합성 기술에 기반한 인 바운드 서비스를 위한 형태로 아웃 바운드 및 팩스 서버 통합을 위한 구조를 제안하고 있지 않다[8-10].

본 논문에서는 W3C에서 현재 수집하고 있는 호 제어 요구사항[4]을 바탕으로 아웃 바운드 서비스를 위한 VoiceXML 플랫폼을 설계하였고, 음성 대화 중에 팩스 송수신을 함께 이용할 수 있도록 VoiceXML 문서형 정의를 확장 설계하였다. 이러한 설계를 바탕으로 팩스 및 아웃 바운드 음성 서비스가 가능한 확장된 VoiceXML 플랫폼을 구현하였다.

II. 확장된 VoiceXML 플랫폼

본 논문에서 설계, 구현한 VoiceXML 플랫폼은 그림 1 과 같이 VoiceXML 서버와 팩스 서버, 그리고 문서 서버로 구성된다.

VoiceXML 서버: 기존의 인 바운드 콜 기반 VoiceXML 인터프리터[9,10]를 팩스 및 아웃 바운드 콜이 가능하도록 확장한 것이다. 크게 자원 관리 쓰레드, 콜 제어 쓰레드, 인터프리터 쓰레드, 음성 인식 쓰레드, 음성 합성 쓰레드로 구성되며 콜 제어 쓰레드는 각 채널별로 생성되어 인 바운드 콜, 아웃 바운드 콜, 팩스 송수신 및 그 수행을 통제한다. 인터프리터 쓰레드는 VoiceXML 1.0 기반 음성 대화 마크업과 음성 대화 중에 인 바운드 및 아웃 바운드 팩스를 이용할 수 있도록 확장한 마크업을 해석하여 사용자와의 음성 대화를 관장한다.

팩스 서버: 크게 팩스의 경우, 전화와 동일하게 적용되며 사용자로부터의 팩스는 인 바운드, 사용자에게 팩스를 전송하는 것은 아웃 바운드로 구분한다. 팩스 서버는 VoiceXML 서버로부터 팩스의 송수신 요청을 받아 이를 수행하는 서버이다.

문서 서버: 음성정보 서비스 응용 별로 작성된 음성 서비스 시나리오에 해당하는 VoiceXML 문서를 생성, 저장하며, VoiceXML 서버의 요청에 따라, 해당하는 문서를 전달한다. 또한 팩스의 송수신에서 발생하는 팩스 이미

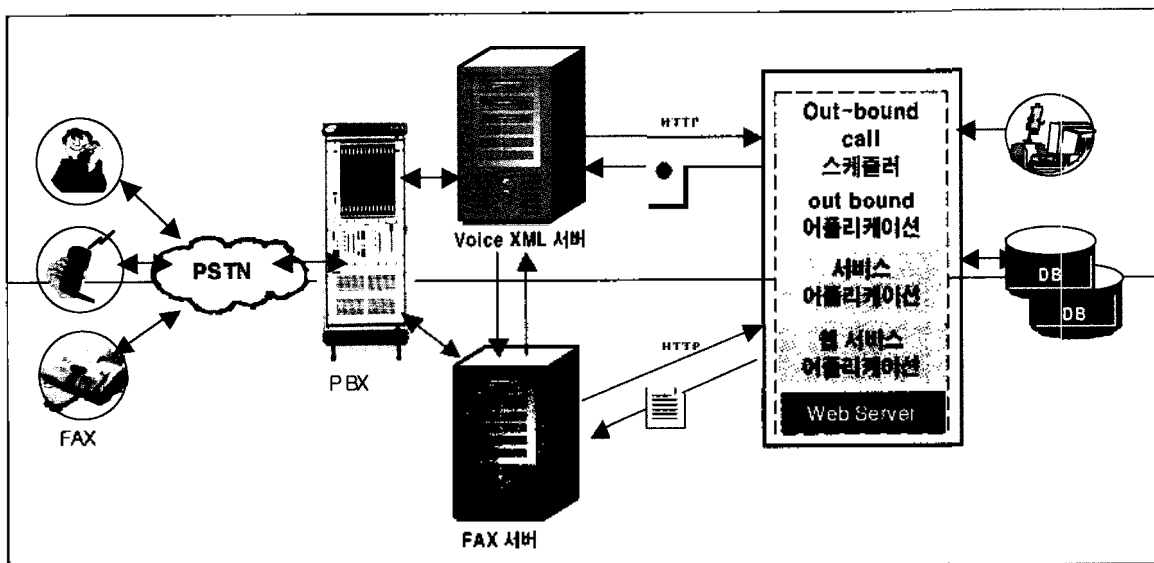


그림 1. 시스템 구조
Fig. 1. System architecture.

지를 저장 관리하는 기능을 포함한다. 특히, Push형 음성 서비스를 위하여 아웃 바운드 스케줄러와 아웃 바운드 음성 서비스를 위한 VoiceXML 문서관리를 포함한다.

III. VoiceXML 기반 Push형 음성 서비스

자동 응답 서비스 등과 같이 인 바운드 서비스의 수행은 채널 단위로 생성된 콜 제어 쓰레드가 채널의 상태를 감시하다가 사용자의 콜 요청 (Inbound call)이 오면 VoiceXML 인터프리터 쓰레드를 생성하여 음성 서비스를 수행한다. 이러한 형태의 서비스는 사용자의 요청이 있어야 서비스가 실행되는 주문형 (on-demand)이다.

VoiceXML 1.0에는 Push형 음성 서비스 (outbound call)

를 위한 엘리먼트가 정의되어 있지 않고, 올해 4월부터 진행 중인 W3C의 호 제어 요구사항에서도 아웃 바운드의 발생 시점 및 활성화 주체에 대해 명시하지 않고 있다. VoiceXML 인터프리터는 VoiceXML 문서를 해석하여 사용자와의 음성 대화를 수행하는 모듈이며, VoiceXML 인터프리터 쓰레드는 콜 제어 쓰레드에 의해서 시작된다. 인 바운드와 아웃 바운드 서비스의 차이는 콜 처리의 차이이며, VoiceXML 인터프리터의 차이는 없다.

인 바운드 음성 서비스는 사용자의 요청에 따라 이루어지므로, VoiceXML 서버의 콜 제어 쓰레드의 콜 감지에 의하여 이루어진다. 그러나 아웃 바운드 음성 서비스의 경우, 기업 등의 정보 시스템에 의하여 이루어지고 아웃 바운드 서비스 논리와 관련 정보는 문서 서버로부터 시작된다. 본 논문에서 설계, 구현한 아웃 바운드 음성 서비스가 가능한 VoiceXML 플랫폼은 그림 2와 같다.

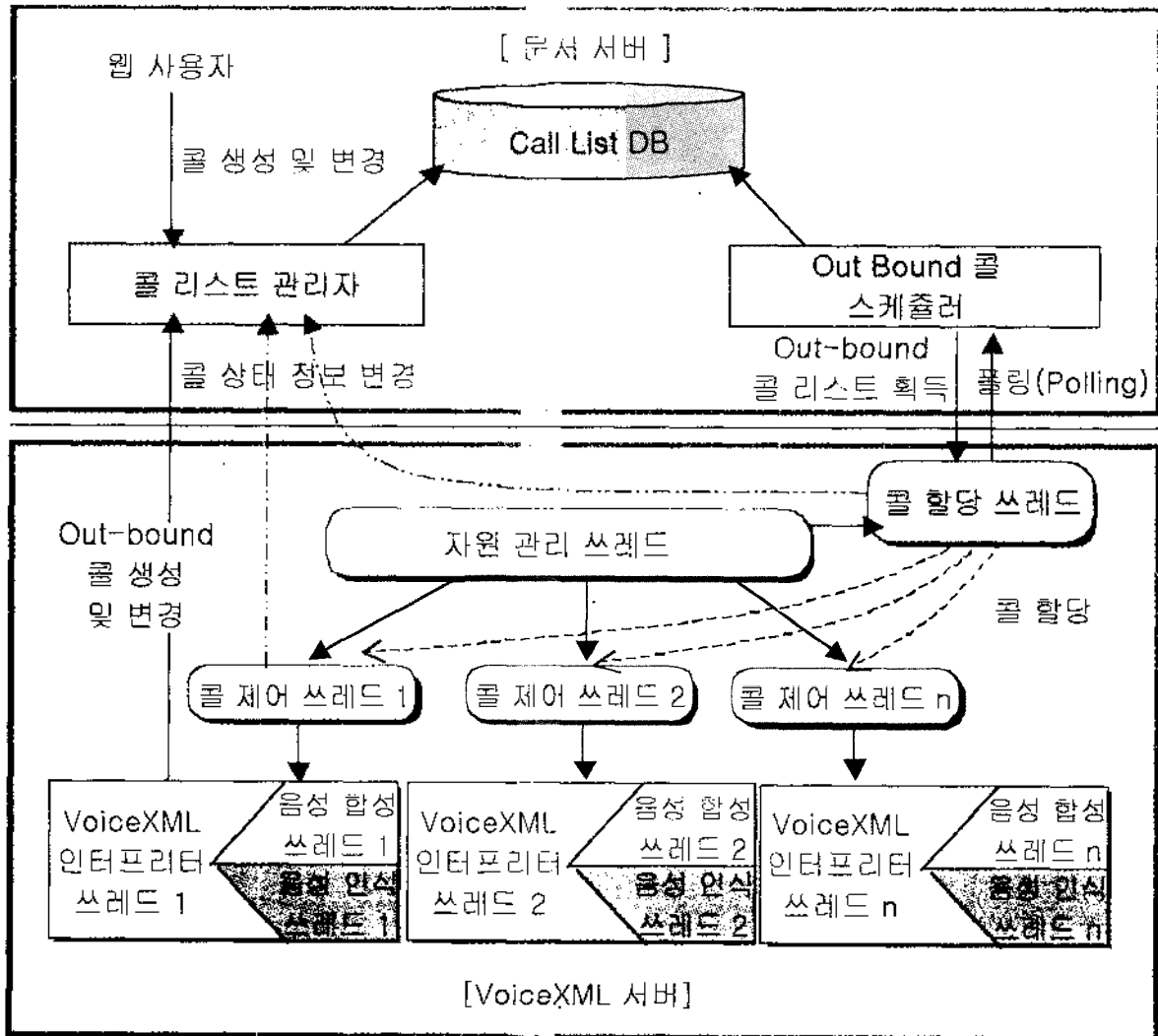


그림 2. 아웃 바운드 서비스를 위한 시스템 구조
Fig. 2. System architecture for outbound call.

표 1. 아웃 바운드 음성 서비스 수행 알고리즘
Table 1. Out bound call service algorithm.

1. 콜 할당 쓰레드는 아웃 바운드 콜 스케줄러에게 아웃 바운드 콜을 수행할 데이터 요청
2. 아웃 바운드 콜 리스트 획득
3. 콜 할당 쓰레드의 채널 할당
4. 콜 시도
 - A. 콜 연결 성공
 - i. 연결 시도에 대한 상태 정보를 문서 서버의 콜 리스트 관리자에게 제공.
 - ii. VoiceXML 문서 요청
 - iii. VoiceXML 문서 획득
 - iv. VoiceXML 인터프리터 쓰레드에 의한 음성 서비스 수행
 - B. 콜 연결 실패
5. 연결 시도에 대한 상태 정보를 문서 서버의 콜 리스트 관리자에게 제공.

아웃 바운드 음성 서비스를 위한 데이터 수집 및 스케줄 관리는 문서 서버 (어플리케이션 서버)에서 수행하고, VoiceXML 서버는 아웃 바운드 콜을 수행할 콜 할당 쓰레드를 생성하여 인터프리터 컨텍스트에 설정된 문서 서버의 아웃 바운드 서비스 스케줄러를 일정 주기로 폴링 (polling)하여 콜 리스트를 획득한다. 콜 할당 쓰레드는 현재 가용 채널인지 채널의 상태를 판단하여 콜을 할당하고 콜이 성공적으로 연결된 경우, 또는 busy, noanswer, network_busy, network_disconnect 등과 같은 오류가 발생한 경우에 따라 그 상태 정보를 문서 서버의 아웃 바운드 서비스 정보원에 반영한다. 콜 상태 정보 (현재 서비스 중에 있음, 연결 실패, 성공적 수행, 초기 상태, 콜 시도 중 등등)는 특히, 다중 서버 환경에서 동일한 콜이 생성되는 것을 방지하기 위해 사용된다.

문서 서버의 스케줄 관리 어플리케이션은 VoiceXML 서버 요청이 올 때마다 우선 순위, 콜 요청 작업 큐에 입력된 시간, 콜 예약 시간, 마지막 콜 시도 시간, 콜 상태, 시도 횟수 등의 요소에 따라 아웃 바운드 콜 리스트를 생성하고 VoiceXML 서버에게 제공한다.

연결에 성공하면, 아웃 바운드 음성 서비스를 위한 VoiceXML 문서 URI (Uniform Resource Identifier)를 문서 서버에 요청하고, VoiceXML 인터프리터 쓰레드를 수행한다. 아웃 바운드 음성 서비스의 종료는 인 바운드의 경우와 동일하다. 이상과 같은 아웃 바운드 음성 서비스 알고리즘은 표 1과 같다.

IV. 팩스 지원 VoiceXML 문서형 정의 확장

VoiceXML 1.0에는 팩스 기능을 정의한 엘리먼트가 없

표 2. <fax> 엘리먼트를 위한 문서형 정의
Table 2. DTD for <fax> element.

```

<!ELEMENT fax (%audio: | %event.handler: |
    filled | grammar | prompt | property)*>
<!ATTLIST fax
    %itemattrs:
    type CDATA #IMPLIED
    beep %boolean: 'false'
    namelist %field.names: #IMPLIED
    app %uri: #IMPLIED
    method (sendreceive) "send"
    bridge %boolean: 'false'
    dest %uri: #IMPLIED
    destexpr %expression: #IMPLIED
    src %uri: #IMPLIED
    srcexpr %expression: #IMPLIED
    textdata CDATA #IMPLIED
    %cache.attrs:
    fetchaudio %uri: #IMPLIED
    
```

대[1]. 그러나 기존 컴퓨터 전화 통합 응용 음성 정보 서비스에서는 주문형 팩스 서비스 (FOD: Fax On Demand), 팩스 메시지 서비스 (FMS: Fax Messaging System)와 같은 팩스를 활용한 다양한 서비스를 제공하고 있고 팩스는 현재 널리 사용중인 통신 수단 중의 하나이다. 또한, 현재 W3C의 호 제어 요구사항에서도 아웃 바운드 팩스에 대한 명세를 기술하고 있다[4]. 표 2와 표 3은 본 논문에서 정의한 <fax> 엘리먼트의 문서형 정의와 속성을 나타낸다. <fax> 엘리먼트는 파일이 첨부된 전자 우편 또는 음성으로 제공할 수 없는 정보를 음성과 함께 제공할 수 있도록 한다.

4.1. 인 바운드 팩스

인 바운드 팩스의 경우 (그림 3), VoiceXML 서버는 팩스

표 3. <fax> 엘리먼트의 어트리뷰트
Table 3. Attributes for <fax> element.

속 성	설 명
o * name	결과를 유사할 다이얼로그 범위의 필드 아이템 (field item) 변수 이름.
o * expr	필드 아이템 (field item) 변수의 초기값. 기본값은 ECMAScript undefined[5]. 초기값이 주어진 경우 필드 아이템 (field item) 변수를 지울 때까지 필드 아이템 (field item)은 방문되지 않음.
o * cond	논리 연산 조건 (Boolean condition). "true"인 경우만 수행.
o * type	팩스 이미지의 MIME 타입. (예) TIFF (Tag Image File Format).
o beep	비프 음 출력 여부. "true"이면, tone이 팩스 수신 전에 발송. 기본값은 "false".
o * namelist	팩스 수행을 위한 데이터 리스트 (예) 송신 요청 시간, 송신 실패 후 후처리 작업 정보.
o * app	아웃 바운드 팩스의 경우, 팩스 송신 요청을 처리하기 위한 어플리케이션 URI. 인 바운드 팩스의 경우, 팩스 이미지를 전송하기 위한 어플리케이션 URI.
o * method	팩스 전송/수신 가능 설정. 기본값은 "send".
o * bridge	"true"인 경우, 팩스 수신/전송이 완료할 때까지 기다림. "false"인 경우, 인 바운드: 콜 라우팅 후, 이벤트 발생. 아웃 바운드: 팩스 전송을 지시하고 다음 단계 진행.
* dest	팩스를 전송할 URI (phone, IP telephony address).
* destexpr	목적지 URI를 생성할 ECMAScript 표현식.
* src	전송할 데이터의 URI.
* srcexpr	전송 데이터 URI를 생성할 ECMAScript 표현식.
* textdata	전송할 텍스트 데이터.
* caching	VoiceXML 1.0 사양 참조
* fetchhint	VoiceXML 1.0 사양 참조
* fetchtimeout	VoiceXML 1.0 사양 참조
* fetchaudio	VoiceXML 1.0 사양 참조

*: 아웃 바운드 팩스에서 사용 o: 인 바운드 팩스에서 사용

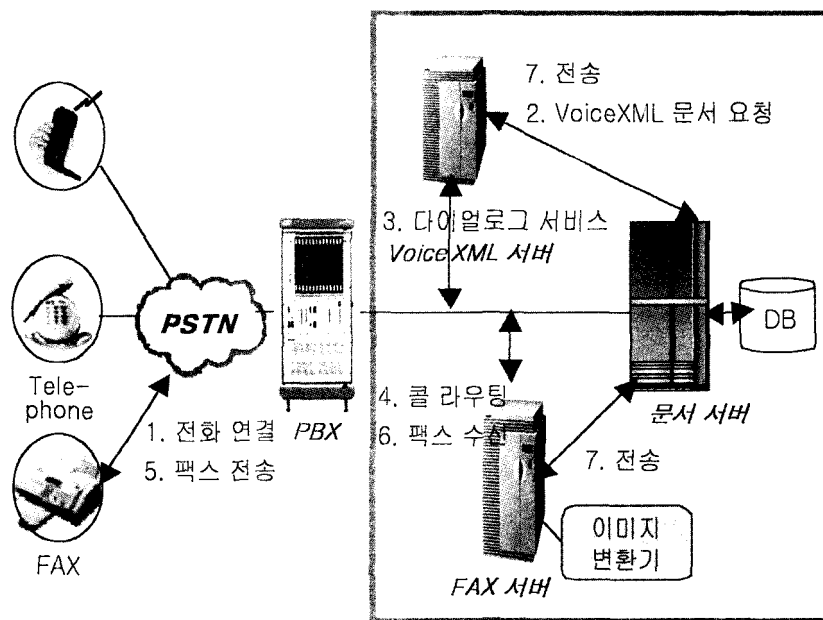


그림 3. 인 바운드 팩스 흐름
Fig. 3. Inbound fax flow.

표 4. 인 바운드 팩스 서비스 알고리즘
Table 4. Inbound fax service algorithm.

1. Inbound 및 Outbound call 연결.
2. 문서 서버에게 서비스 시나리오, VoiceXML 문서 요청
3. 2 에서 획득한 VoiceXML 문서를 이용하여 음성 서비스 수행.
4. VoiceXML 인터프리터는 음성 서비스 수행 중에 <fax> 엘리먼트를 만나면 method 어트리뷰트를 검사하여 그 값이 "receive"인 경우, 팩스 수신을 위해 팩스 서버에게 콜 라우팅을 시도.
 - A. 콜 라우팅에 실패한 경우
 - i. 시스템 환경 (인터프리터 컨텍스트)에 설정된 값 만큼 반복적으로 시도
 - B. 콜 라우팅에 성공한 경우
 - i. fax 엘리먼트의 bridge 어트리뷰트 값이 "false"인 경우, telephone.disconnect 이벤트를 발생하고, "true"인 경우, 팩스 수신을 완료할 때까지 인터프리터 대기.
 - ii. 사용자는 서비스 시나리오에 따라 안내 멘트 및 비프음을 듣고 팩스 전송.
 - iii. 팩스 문서 수신.
 - iv. 팩스 서버는 수신된 문서를 VoiceXML 시나리오 구성에 따라 문서 서버로 직접 전송하거나 (fax 엘리먼트의 app 어트리뷰트에 명시된 경우), fax 엘리먼트의 name 어트리뷰트에 저장. * fax 엘리먼트의 name 어트리뷰트는 submit 엘리먼트를 통한 데이터 전송에 이용.
 - v. fax 엘리먼트의 bridge 어트리뷰트 값이 "true"인 경우, 인터프리터로 복귀하여 남은 서비스 수행.

표 5. 인 바운드 팩스 서비스를 위한 VoiceXML 예제
Table 5. VoiceXML example of inbound fax service.

```

<?xml version = "1.0" encoding = "euc-kr"?>
<vxml version = "1.0">
  <form>
    <fax name = "mycall" beep = "true" bridge = "true" method = "receive"
      type = "tiff">
      <prompt> 삐 소리가 나면 팩스를 넣어주세요 </prompt>
      <filled>
        <if cond = "mycall == 'busy' ">
          <prompt> 팩스 서비스를 이용할 수 없습니다. </prompt>
          <exit/>
        <else/>
          <submit next = "http://krkim/vxml/faxprocessing.asp" namelist = "mycall"
            enctype = "multipart/form-data" method = "post"/>
        </if>
      </filled>
    </fax>
  </form>
</vxml>

```

서버에게 팩스 수신을 지시하고 bridge 속성 값에 따라 VoiceXML 인터프리터 수행을 제한한다. bridge 값이 거짓이면 팩스 서버와 연결된 후 telephone.disconnect 이벤트를 발생하여 지금까지의 음성 서비스를 종료하고, bridge 값이 참이면 VoiceXML 인터프리터는 팩스 서버의 팩스 수신 처리를 완료할 때까지 기다린 후 음성 대화 서비스를 계속한다. 팩스 문서는 TIFF 형태의 그림 파일로 전송되며 app 속성 또는 <submit> 엘리먼트를 통해 문서 서버로 전송된다.

표 4는 인 바운드 팩스의 수행 알고리즘을 나타내고, 표 5는 인 바운드 팩스 서비스를 위한 VoiceXML 문서의 한 예이다.

4.2. <submit> 통한 팩스 데이터 전송

VoiceXML은 HTTP를 기반으로 음성 서비스 과정에서 획득한 사용자 입력을 문서 서버로 전송하거나 서비스 수행에 필요한 데이터를 문서 서버에게 요청한다. <submit>은 <fax>, <record> (음성 녹음을 위한 엘리먼트), <field>

표 6. 팩스를 위한 <submit> 엘리먼트
Table 6. <submit> element for fax data.

```

<submit next = "http://krkim/vxml/faxprocessing.asp" namelist = "mycall"
        enctype = "multipart/form-data" method = "post"/>

<Header section>
MIME-Version: 1.0\r\nContent-Type: multipart/form-data; boundary=-----7fs9dfsf9sdf

<Body section>
-----7fs9dfsf9sdf
Content-Disposition: form-data; name = "FILE1"; filename = "mycall.tiff"
Content-Type: application/octet-stream
... contents of mycall.tiff...
-----7fs9dfsf9sdf-
    
```

등의 수행 후 생성된 이미지 파일, 오디오 파일, 텍스트 등을 문서 서버로 전송하기 위한 방법으로 문서 수행의 결과 값을 다른 VoiceXML 문서에 반영할 수 있도록 한다. 표 6은 <fax> 엘리먼트를 수행하여 생성된 이미지 파일을 전송하기 위한 "post" 방식의 <submit> 엘리먼트 예제이다. <submit>는 VoiceXML 1.0 사양에 포함되어 있다.

속성 값인 next에는 전송 데이터를 처리하기 위한 웹 어플리케이션 URL이 지정되고 namelist 속성을 통해 전송할 데이터를 열거한다.

method 속성은 전송 방법 즉, "get" 또는 "post"를 나타내는데, VoiceXML은 기본 값으로 "get" 방식을 사용한다. 그러나, "get" 방식은 전송할 수 있는 데이터의 크기가 서버 의존적이고 크기에 제한이 있어 많은 양의 데이터 전송에 부적합하므로 <fax>나 <record>에서 생성된 이미지 및 오디오 파일을 전송하기 위해서는 "post" 방식을 사용해야 한다.

enctype은 MIME (Multipurpose Internet Mail Extensions) 타입을 나타내며, 기본값으로 "application/x-www-form-urlencoded"이 지정되나 많은 양의 이진수(binary)데이터 또는 비아스키(Non-ASCII) 글자를 포함하는 텍스트에서는 비효과적이므로 파일, 비아스키 데이터, 이진수 데이터를 포함하는 팩스 데이터는 multipart MIME 타입으로 지정해야 한다. 따라서, namelist에 <fax>, <record>의 form item 변수를 포함한 경우 multipart MIME 타입으로 지정해야 한다. 전송 데이터의 크기에 제한이 없는 post 방식을 사용하여 텍스트, 이미지, 오디오 데이터를 전송하기 위해서는 콘텐츠 타입(MIME type)을 포함하는 엔티드 헤더와 실제 데이터를 포함하는 엔티드 바디를 제공해야 한다[7]. 이상의 팩스를 위한 데이터 송수신 자료 구조는 표 7과 같다.

표 7. Post 방식의 데이터 전송 형식
Table 7. Data transmission format for post method.

```

<Header section>
header name: header value

<Body section>
--boundary string
field section 1
--boundary string
field section 2
--boundary string--

<Field section>
Content-Disposition header line
Empty line
Body section line
    
```

4.3. 아웃 바운드 팩스

아웃 바운드 팩스는 사용자에게 팩스를 전송하는 것으로, textdata 또는 src에 명시된 데이터를 팩스 합성하여 dest 또는 destexpr로 지정된 목적지에 전송한다. 팩스 합성이란 데이터를 팩스 형식, 예를 들어, TIFF나 RAW 형태의 이미지로 변환시키는 기능을 말한다. VoiceXML 서버는 팩스 수행에 대한 정보를 문서 서버의 한 응용에 제공하고 팩스 서버는 문서 서버로부터 이 정보를 획득하여 아웃 바운드 팩스 서비스를 수행한다. 또한 bridge 속성을 이용하여 참인 경우, 팩스 전송을 완료한 후 VoiceXML 인터프리터 수행을 계행하고, 거짓인 경우 전송 지시만을 전달하고 VoiceXML 서비스를 계속 수행한다. 그림 4는 아웃 바운드 팩스 흐름을 나타낸다.

표 8은 아웃 바운드 팩스의 수행 알고리즘을 나타내고, 표 9는 아웃 바운드 팩스 서비스를 위한 VoiceXML 시나리오 문서 예제이다.

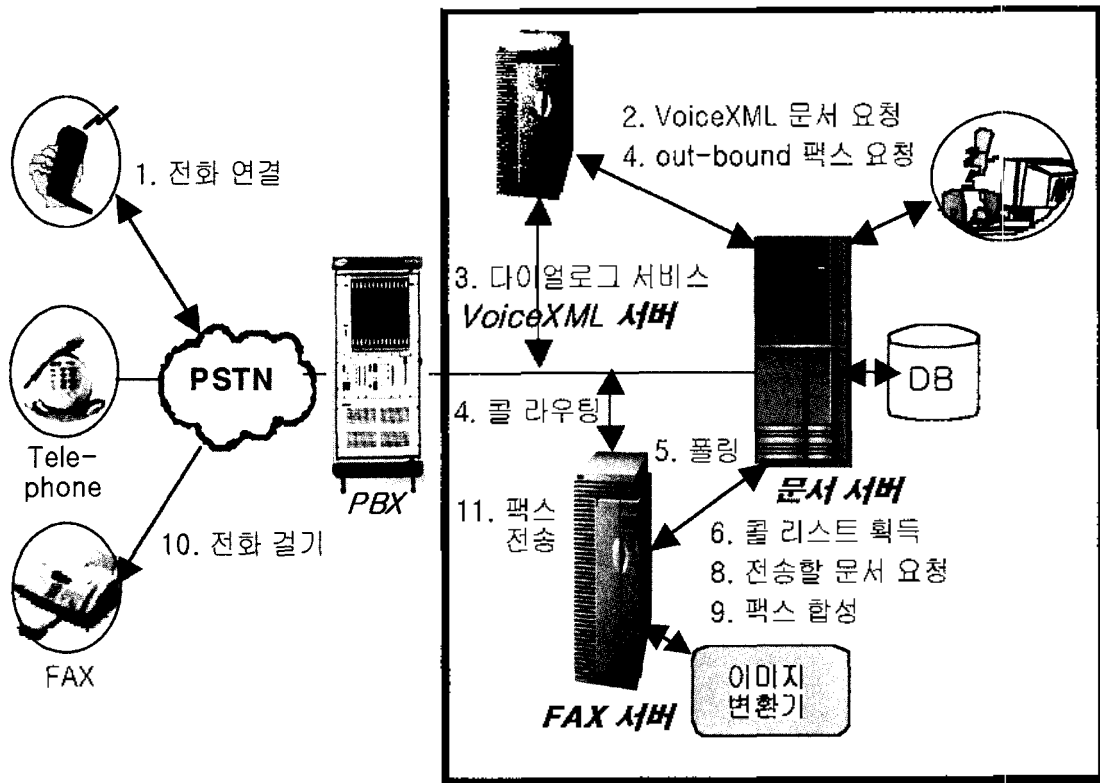


그림 4. 아웃 바운드 팩스 흐름
Fig. 4. Outbound fax flow.

표 8. 아웃 바운드 팩스 서비스 알고리즘
Table 8. Outbound fax service algorithm.

1. 인 바운드 팩스의 1, 2, 3 단계와 유사.
2. VoiceXML 인터프리터는 서비스 수행 중에 <fax> 엘리먼트를 만나면 method 어트리뷰트를 검사하여 값이 "send"인 경우, 팩스 전송을 위해 팩스 서버 또는 문서 서버에게 요청.
 - A. bridge 값이 "false"인 경우,
 - i. app 어트리뷰트에 나타난 어플리케이션에게 팩스 전송에 필요한 데이터 (목적지 정보 및 namelist 에서 설정한 데이터)를 제공.
 - ii. VoiceXML 인터프리터는 팩스 전송이 완료될 때까지 기다리지 않고 남은 서비스 수행.
 - B. bridge 값이 "true"인 경우,
 - i. 팩스 서버에게 팩스 전송을 지시하고 VoiceXML 인터프리터 대기.
 - ii. 팩스 서버의 콜 시도
 - iii. 콜이 성공적으로 연결된 경우, 팩스를 전송하고 팩스 전송이 완료되면 인터프리터로 복귀하여 나머지 서비스 수행.
3. 팩스 서버는 문서 서버의 아웃 바운드 팩스 스케줄러를 일정 주기로 폴링
4. out bound Fax 서비스를 위한 콜 리스트 획득.
(팩스 서버는 아웃 바운드 콜을 위한 시스템과 유사)
5. 팩스 서버는 4 에서 획득한 정보를 이용하여 문서 서버에게 팩스 전송을 위한 자원 요청.
6. 팩스 서버는 요청된 정보를 팩스 전송에 필요한 형식 (예를들어, TIFF)으로 변환.
7. 4 에서 획득한 정보 (팩스 번호)를 이용하여 연결 시도.
 - A. 콜 연결 성공
 - i. 콜 상태 정보 변경 - 연결 시도에 대한 상태 정보를 문서 서버의 팩스 서비스 어플리케이션에게 제공.
 - ii. 팩스 문서 전송
 - B. 콜 연결 실패
 - i. 콜 상태 정보 변경- 연결 시도에 대한 상태 정보를 문서 서버의 팩스 서비스 어플리케이션에게 제공.
8. 연결 시도에 대한 상태 정보를 문서 서버의 팩스 서비스 어플리케이션에게 제공.

표 9. 아웃 바운드 팩스 서비스를 위한 VoiceXML 예제
Table 9. VoiceXML example of outbound fax service.

```

<?xml version = "1.0" encoding = "euc-kr"?>
<vxml version = "1.0">
  <form>
    <block>
      <prompt> 인녕하세요 **님. 팩스 서비스 입니다. </prompt>
    </block>
    <field name = "reservation_time" type = "time">
      <prompt> 팩스 전송 예약시간을 입력해 주세요
          예를 들어, 오후 2시 30분은 1430 입니다. </prompt>
    </field>
    <field name = "user_phone_num" type = "phone">
      <prompt> 지역번호를 포함한 팩스 번호를 입력해 주세요. </prompt>
    </field>
    <field name = "confirm" type = "boolean">
      <prompt> 입력하신 정보 <value expr = "user_phone_num"/>,
          <value expr = "reservation_time"/> 이 맞습니까?
    </prompt>
    <filled>
      <if cond = "confirm == false">
        <clear namelist = "reservation_time user_phone_num confirm"/>
      </if>
    </filled>
  </field>
  <fax name = "traffic" destexpr = "'phone:/' + user_phone_num"
      type = "tiff" src = "http://krkim/traffic.asp"
      app = "http://krkim/requeset_save.asp"
      namelist = "reservation_time session.telephone.ani" bridge = "false"/>
</form>
</vxml>

```

V. 실험결과

본 연구에서 설계, 구현한 VoiceXML 플랫폼 및 VoiceXML 인터프리터는 윈도우 NT/2000에서 운용되며 HTTP 표준 프로토콜을 사용하여 IIS, Apache 등의 다양한 웹 서버를 지원한다. ECMA 스크립트 기능을 구현하기 위해 마이크로소프트사의 MS 스크립트 컨트롤을 사용하였다. 구현된 VoiceXML 인터프리터는 <script> 엘리먼트를 사용하여 정의한 스크립트문에서의 변수와 <var>와 같은 VoiceXML 시나리오에서 선언한 변수가 심볼 테이블을 공유하지 않기 때문에 함수 호출을 통한 변수 바인딩과 같은 제한된 방법으로 지원한다. 또한, 본 연구에서 구현한 VoiceXML 플랫폼은 음성 서비스를 위한 VoiceXML 서버의 부하를 줄이기 위해 독립된 형태로 팩스 서버를 구성하였다.

VI. 결론 및 향후 연구과제

웹, 음성 처리, 컴퓨터 전화 통합 기술을 결합해 새로운 부가 가치를 창출할 수 있는 VoiceXML이 새로운 화두로 등장하고 있다. 또한, 음성 정보 서비스가 복잡하고 다양해지면서 자동 통지 시스템, 무인 자동 교환 안내, 팩스 정보 시스템 등과 같은 서비스가 음성 대화와 함께 요구된다. 특히, 고객관계관리에서는 Push형 음성 서비스가 매우 중요하다. 이와 같은 복합 서비스를 위해서는 현재 VoiceXML 1.0 사양에는 정의되지 않은 확장된 호 제어 기능이 요구된다.

본 논문은 Push형 음성 서비스가 가능한 VoiceXML 플랫폼을 설계 구현하였다. 또한 W3C의 호 제어 요구사항을 바탕으로 <fax> 엘리먼트를 정의하고 VoiceXML 1.0 사양을 확장하였으며, HTTP 기반 기술을 사용한 문서 서버로의 팩스 데이터 전송 방법에 대해 설계하고 구현하였

다. 향후 연구 과제로는 호 제어 메커니즘을 확장하여 컨퍼런스 콜 (conference call) 및 멀티 컨퍼런스 (multi-part conference) 기능을 지원하는 컨퍼런스 콜 생성, 참여, 종료, 제어를 위한 모델을 연구하고 조건적 콜 응답, 대화 일시 중지, 계행 등과 같은 고급화된 제어 메커니즘을 구현하는 것이다. 또한, 문자 인식 (OCR: Optical Character Reader) 기술을 이용하여 팩스 및 웹에서 전송된 이미지 데이터를 읽어주는 기능과 현재 진행중인 VoiceXML 2.0 사양을 지원하는 것이다.

참고 문헌

1. W3C, "Voice Extensible Markup Language VoiceXML" (Version 1.0), 5 May 2000, <http://www.w3.org/TR/voicexml>
2. W3C, "Extensible Markup Language(XML) 1.0" 6 October 2000, <http://www.xml.com/axml/testaxml.htm>
3. W3C, "Voice Browsers," 4 December 2000, <http://www.w3.org/TR/voice-intro>
4. W3C, "Call Control Requirements in a Voice Browser Framework," 13 April 2001 <http://www.w3.org/TR/call-control-reqs/>
5. International Organization for Standardization, "ECMAScript Language Specification" ECMA-262, <http://www.ecma.ch>, Ed.3, December, 1999.
6. Bob Edgar, "The VoiceXML Handbook", CMP, 2001.
7. IETF, "Form-based File Upload in HTML," <http://www.ietf.org/rfc/rfc1867.txt>, November, 1995.

8. 신현경 외, "음성 웹서비스를 위한 VoiceXML 해석기의 설계 및 구현", 한국음향학회지, 제20권 제4호, pp. 42-47, 2001.
9. 김경란, 홍기형, "VXML 편집기와 음성 브라우저의 설계 및 구현," 정보과학회 춘계학술대회, 제27권 제1호 pp. 414-416, 2000.
10. 김경란, 홍기형, "다중 사용자 지원 VoiceXML 서버 설계 및 구현," 한국음향학회 하계학술대회, 제20권 제 호, pp. 175-178, 2001.

저자 약력

• 김 경 란 (Kyoung-Ran Kim)



1999년 2월: 성신여자대학교 전산학과 졸업 (이학사)
 2001년 2월: 성신여자대학교 대학원 전산학과 졸업 (이학 석사)
 2000년 ~ 현재: 미디어포드
 ※ 주관심분야: VoiceXML, 음성 응용, CTI, 음성 처리, Component S/W, XML

• 홍 기 형 (Ki-Hyung Hong)



1985년 2월: 서울대학교 공과대학 전자계산기공학과 졸업 (공학사)
 1987년 2월: 한국과학기술원 전산학과 졸업 (공학 석사)
 1994년 2월: 한국과학기술원 전산학과 졸업 (공학 박사)
 1991년 ~ 1994년: (주)컴퓨터응용기술, 개발실장
 1994년 ~ 1998년: 한국전자통신연구원, 선임연구원
 1998년 ~ 현재: 성신여자대학교 컴퓨터정보학부 조교수
 ※ 주관심분야: VoiceXML, 음성 미들웨어, CTI, ITI, XML, Component S/W, Database Systems