

대량의 발신 호를 지원하는 음성 메시지 시스템*

김정곤(한세대)

<차 례>

- | | |
|----------------------|----------------------|
| 1. 서론 | 3.1. MOCASS 구축 요구 사항 |
| 2. 기존의 음성 메시지 발송 시스템 | 3.2. 제안된 시스템의 구성 |
| 2.1. 기존의 IVR | 3.3. 구현 예제 |
| 2.2. 기존의 MOCASS | 4. 시범 운영 사례 |
| 3. 제안된 MOCASS의 설계 | 5. 결론 및 향후 과제 |

<Abstract>

Voice Message System Supporting Massive Outbound Call

Jeonggon Kim

In this paper, new voice message system supporting massive outbound call is proposed. Basic idea of the proposed system is to pre-process all the text-to-speech conversion process, mixing of text and attached music file and to store the results of pre-process in the cache server which is connected to the IVR. New voice message system is optimized for the voice message system supporting massive outbound call by distributing the load of the web server caused by server-side script implementation which is accessing database and generating dynamic Voice XML document over client module and server module of web server. The proposed voice message system was test-deployed in one domestic voice message application service provider and it is shown that proposed voice message system reduced the response latency problem of test-bed voice message system.

* Keywords: VoiceXML, IVR, Voice Message System, Latency.

* 본 연구는 (주)미디어포드의 지원으로 완성되었으며 논문 연구에 도움을 주신 미디어포드의 김상오 연구소장, 정상윤 선임연구원에게 감사 드립니다.

1. 서 론

VoiceXML (Voice eXtensible Markup Language)[1]은 음성인식, 음성합성, 음성 입력과 DTMF (Dual Tone Multi-Frequency) 입력, 음성 녹음 그리고 전화 통신과 상호 주도 대화 등의 요소 기술을 지원하는 음성 대화 기술 언어로서 Voice XML을 음성 메시지 시스템에 적용하는 목적은 첫째, 표준화된 음성 기술을 사용하여 대화형 음성 언어 인터페이스를 제공하는 표준화된 음성 시나리오를 제공하고 둘째, 대화 시나리오와 같은 응용 프로그램의 개발과 운영을 하위레벨 프로그래밍이나 리소스 관리로부터 분리하여 콘텐츠 제공자는 VoiceXML 표준에 따라 음성 입출력이 가능한 음성 대화 시나리오의 개발에만 전념하고 음성 메시지 서비스 제공자는 음성 대화 서비스 시나리오에 독립되게 장비를 구축할 수 있음으로 다양한 음성 서비스를 신속하게 개발하기 위한 것이라고 할 수 있다[2].

이러한 VoiceXML의 장점을 이용하여 구축할 수 있는 대표적인 음성 메시지 서비스 응용분야는 VMSP (Voice Message Service Provider), 고객 콜센터, 뉴스, 일기, 스포츠, 증권, 교통 정보를 제공하는 음성 정보 서비스 등이며 이들 응용 분야들 중에서 불특정 다수에게 음성 메시지를 제공하거나 (재난 통보의 경우나 홍보성 메시지) DB (Database)와 VoiceXML과의 연동을 통하여 특정 다수에게 개인화된 발신 음성 메시지(Personalized Outbound Voice Message)를 전달하는 서비스가 단문 메시지(SMS: Short Message Service) 시장을 대체하여 많은 주목을 받을 것으로 예측된다.

또한 비즈니스 모델 측면에서 보았을 때 음성 발신 메시지를 발송하는 사업자는 고비용의 음성 메시지 시스템을 독립적으로 소유하지 않고 음성 메시지 서비스를 제공하는 응용 프로그램 서비스 사업자(Application Service Provider)의 시설이나 서비스를 임대하여 사용한다면 경비 절감과 콘텐츠 개발에 전념할 수 있는 이익을 누리게 될 것이므로 음성 메시지를 전달하는 응용 서비스 제공자는 음성 정보 시장에서 많은 수요를 창출 할 것으로 예상된다.

위에서 언급한 두 가지 특성을 갖춘 음성 메시지 서비스 시스템은 음성 메시지 응용프로그램 서비스 제공자가 다수의 음성 메시지 발신자로부터 음성 메시지 발송을 요청 받아서 일괄적으로 처리하는 시스템으로서 사업자는 효율적으로 대량의 발신 호를 지원하는 응용 서비스 시스템 (MOCASS: Massive Outbound Call Application Service System)을 구축할 필요가 있다.

기존의 음성 메시지 발송 시스템은 음성 메시지 발송자가 웹 서버(Web Server)에 입력한 정보를 ASP (Active Server Pages)나 CGI (Common Gateway Interface)와 같은 인터페이스를 통하여 서버사이드 스크립트(Server-side Script)를 사용하여 VoiceXML 기반의 음성 메시지를 동적으로 생성하고 이 메시지를 VoiceXML 인터프리터를 내장한 IVR (Interactive Voice Response)이 발송하는 시스템으로서 이러한

기존의 시스템은 대량의 호를 발신하는 경우에 서버와 TTS (Text-To-Speech) 엔진의 부하가 증가하고 응답이 지연(Latency) 되는 문제점을 가지고 있다[3].

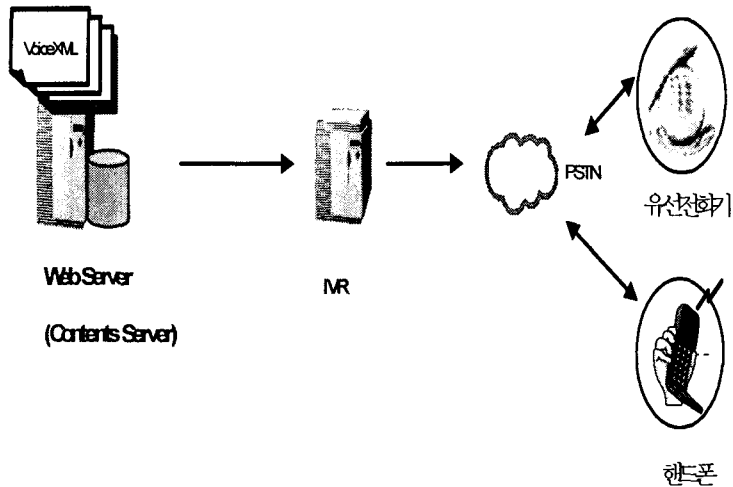
본 논문에서는 이러한 문제점을 개선하기 위하여 대량의 발신 호를 효율적으로 지원할 수 있는 새로운 시스템(MOCASS)을 제안하였다. 제시한 시스템은 텍스트의 음성 변환, 변환된 음성과 배경 음악과의 합성을 호 발송 전에 전처리(pre-process)하고 음성메시지 실행 때에 필요한 오디오 파일을 호 발송 전에 사전에 캐시 서버에 저장하여 IVR이 VoiceXML 문서 실행에 필요한 오디오 파일을 실시간으로 접근할 수 있도록 하여 응답이 지연되는 것을 개선하였으며 효율적인 전처리를 위하여 웹 서버의 기능을 내부에서 클라이언트 모듈과 서버 모듈로 분리하였다.

본 논문의 구성은 다음과 같다. 2장에서는 기존의 음성 메시지 발송 시스템에 대하여 설명하였다. 3장에서는 새롭게 제안하는 시스템의 요구 사항과 구성 요소의 기능에 대하여 설명하고 제안하는 시스템에서 음성 메시지가 생성되는 과정에 대하여 설명하였다. 4장에서는 제안하는 시스템의 시범 운영 사례에 대하여 설명하였고 마지막으로 5장에서는 결론을 맺는다.

2. 기존의 음성 메시지 발송 시스템

2.1. 기존의 IVR

<그림 1>은 VoiceXML 기반의 IVR을 사용하여 구축한 기존의 음성 메시지 시스템을 보여 준다. <그림 1>의 음성 메시지 시스템에서는 IVR과 웹 서버(또는 콘텐츠 서버)가 클라이언트(Client)와 서버(Server)의 관계를 이루면서 IVR이 PSTN (Public Switched Telephone Network)을 통하여 유선전화기나 이동전화기를 보유한 다수의 사람들에게 전화로 연결고 인터넷을 통하여 음성 메시지 발신자의 콘텐츠 서버로부터 VoiceXML 문서(시나리오)를 가지고 와서 음성 메시지를 전달하게 된다.

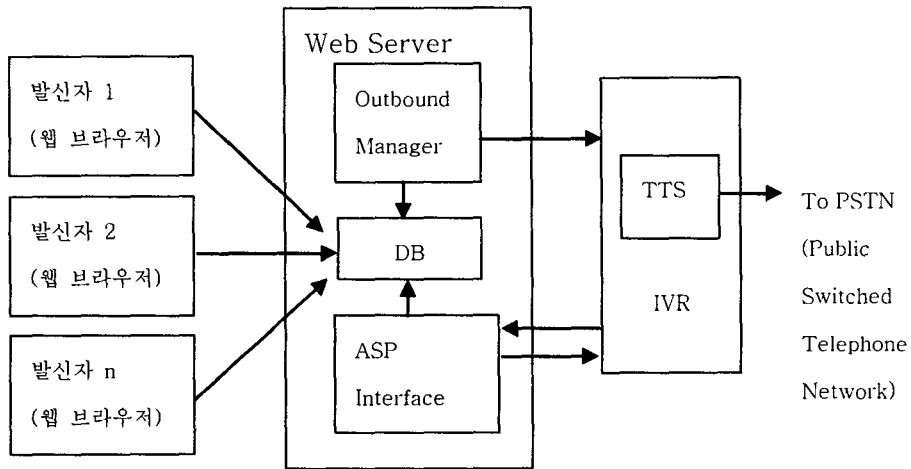


<그림 1> IVR의 클라이언트 - 서버 구성

이 경우에 IVR은 VoiceXML 인터프리터를 구동하여 VoiceXML 기반의 시나리오를 해석하고 실행하게 되며 발신자가 입력한 텍스트를 TTS 엔진을 이용하여 음성으로 변환하거나 발신자가 녹음한 오디오 파일의 음성 메시지를 수신자에게 들려주어서 음성 메시지를 수신자에게 전달하게 된다. 또한, 발신자의 요청에 따라 텍스트 메시지와 배경음악을 동시에 제공하여야 하는 경우에는 IVR은 TTS를 사용하여 텍스트를 변환한 음성과 배경음악을 합성하는 기능을 제공한다.

2.2. 기존의 MOCASS

<그림 1>의 시스템 구성을 확장하여 구성한 기존의 대량의 음성 메시지 발송 서비스 시스템(MOCASS)은 <그림 2>와 같다. <그림 2>는 다수의 음성 메시지 발신자가 MOCASS 사업자가 제공하는 인터넷 웹 인터페이스를 통하여 음성메시지 발신 서비스를 요청하면 사업자는 음성 메시지를 발송하고 서비스 사용료를 청구하는 응용 프로그램 임대 서비스(Application Service Provider)를 나타낸 것이며 웹 서버와 IVR이 Windows 운영체제와 ASP (Active Server Pages) 인터페이스를 사용하는 시스템을 가정하였다.



<그림 2> 기존의 MOCASS의 구성도

위의 시스템의 음성 메시지 발신 서비스의 흐름은 다음과 같다.

음성 메시지 발송자:

- 1) 음성 메시지 발송자는 MOCASS 사업자가 제공하는 인터넷 웹 사이트를 방문하여 발신자, 수신자, 음성메시지, 발송 시간, 발송 실패시의 대응 방안 등을 입력하고
- 2) 원하는 템플레이트(template) (예를 들어, 연체 통지, 생일 파티 초대 등)를 지정하면 이러한 정보들은 웹 서버의 DB에 저장된다.

OM (Outbound Manager):

- 3) OM (Outbound Manager)는 웹 서버의 DB를 정기적으로 조회하여 음성 메시지 발송이 필요한 경우에는 IVR에게 수신자 전화번호와 작업 코드 (Job Code)를 전달하고 음성 메시지 발송을 지시한다.

IVR:

- 4) OM의 지시를 받은 IVR은 수신자에게 전화를 걸고
- 5) (수신자가 전화를 받으면) 웹 서버의 ASP (Active Server Pages)로 작성된 VoiceXML 기본 문서(root document)를 접근하여 작업 코드를 넘겨주고

웹 서버:

- 6) 웹 서버는 작업 코드에서 지정한 데이터 (음성 메시지 중에서 발신자가 입력한 부분을 포함하여 VoiceXML 문서 생성에 필요한 자료)를 DB에서 선택(Select)하고
- 7) 위의 6)에서 선택한 데이터를 기반으로 VoiceXML 문서를 생성하여 IVR에게 전달하고

IVR:

- 8) IVR은 웹 서버로부터 받은 VoiceXML 문서를 실행하게 된다.

이와 같이 웹 서버가 웹 서버 내부의 DB와 응용 프로그램인 IVR을 동적으로 결합(dynamic binding)시키는 방법은 콘텐츠를 동적으로 생성할 수 있는 장점은 있으나 위와 같은 구성으로 대량의 발신 호(Massive Outbound Call)를 지원할 수 있는 효율적인 시스템을 구성하는 데에는 다음과 같은 문제점이 있다.

- 1) 음성 메시지 발송을 처리하기 위하여 웹 서버가 서버 사이드 스크립트를 수행하는 동안 웹 서버에 부하가 증가하고 클라이언트인 IVR이 서버를 기다리는 동안 서비스가 지연된다.
- 2) IVR이 대량의 발신 호를 동시에 처리하는 경우에는 IVR이 시나리오에 포함된 텍스트를 오디오 파일로 변환하기 위하여 연속적으로 TTS를 구동하고 TTS에 많은 양의 부하가 (거의) 동시에 작용하여 수신자에 대한 IVR의 응답 지연(Latency)이 증가하고 전체 시스템의 성능이 저하되는 문제점을 안고 있다. 또한 음성 메시지 발송자가 요구하는 개인화된 서비스를 제공하기 위하여 MOCASS가 유사한 메시지를 다수의 수신자에게 반복하여 발송하는 경우에 수신자의 이름이나 일부 가변하는 메시지를 제외하고 같은 내용의 메시지를 반복하면서 연속적으로 TTS엔진을 구동하게 되어 TTS의 응답속도가 느려지고 결과적으로 IVR과 음성 메시지 발송 시스템 전체의 성능이 저하하는 문제점을 가지고 있다[3].

일반적으로 웹 서버나 IVR의 성능을 향상시키는 방법으로서 IVR이 웹 서버에서 받은 콘텐츠를 일정 시간 동안 IVR 내부에 저장하는 캐시 기술(caching)을 적용하는 것은 잘 알려져 있으나 MOCASS와 같이 콘텐츠가 동적으로 생성되는 경우에는 캐시 기술을 사용하여 응답 지연을 해결하기는 어렵다[4][5]. 또한, Tellme.com 등과 같은 음성 포털 사이트에서는 캐시 기술을 활용하기 위하여 IVR이 정적인 콘텐츠와 동적으로 생성되는 데이터를 분리하여 정적인 콘텐츠 부분만을 IVR 내부의 캐시에 저장하고 IVR이 웹 서버의 CGI를 통하여 동적으로 생성되

는 부분만을 XML (eXtensible Markup Language) 문서로 받아서 처리하여 서버의 부담을 줄이는 방법을 제안하였고[6] widerthan.com에서는 IVR이 CGI를 거치지 않고 JDBC (Java Database Connectivity)인터페이스를 통하여 직접 DB를 접속하여 VoiceXML 문서를 IVR의 캐시에 저장하는 방법을 제안하여 응답속도가 개선되고 웹 서버의 부하를 줄일 수 있는 방법을 제안하였지만[5][7] 대량의 발신 호를 동시에 처리하는 경우와 같이 연속적으로 TTS를 구동할 때 발생할 수 있는 응답 지연의 문제는 여전히 남아 있다고 할 수 있다.

그러나 응답 시간의 지연을 줄이기 위하여 IVR이 수행하고자 하는 VoiceXML 문서의 외부에 연결되어 있는 문서나 문법(grammar), 데이터 등을 전처리(Pre-process)하는 방법이나[3] TTS가 텍스트를 변환한 오디오 파일을 캐시 서버(또는 IVR의 하드 디스크)에 미리 저장하는 방법을 제안하여[8] 위와 같은 문제점을 개선할 수 있는 가능성을 제시한 연구 결과도 발표되었다.

3. 제안된 MOCASS의 설계

3.1. MOCASS 구축 요구 사항

본 논문에서는 위의 문제점을 개선하여 효율적인 음성 메시지 서비스를 구축할 수 있는 새로운 형태의 MOCASS 구축 시에 필요한 요구 사항을 다음과 같이 정의하였다.

첫째, MOCASS 사업자가 다양한 음성 메시지 발송자를 수용하기 위하여 전체 시스템은 분산 시스템으로 구축되어야 한다. 여기서 분산 시스템은 MOCASS는 중앙에 음성 메시지를 발송하는 서버를 두며 음성 메시지 발신자는 클라이언트 모듈을 통하여 음성 메시지 발신에 필요한 최소한의 가변적인 정보만을 서버에 전송하고 메시지 전송에 필요한 고정적인 정보(예를 들면, 템플레이트)는 서버에 저장한다. 이 요구 사항은 음성 메시지 발송자가 음성 메시지 발신 전용 클라이언트 프로그램을 사용하거나 인터넷 브라우저를 통하여 서버에 접속하고 음성 메시지 발신에 필요한 최소한의 가변적인 정보만을 서버에 전송하는 방법을 사용할 수 있다.

둘째, 음성 메시지를 발송하는 시스템은 동일 또는 유사한 메시지를 대량으로 발송하는 데에 적합하고 효율적인 시스템이어야 한다. 예를 들면, 국가의 재난 상황을 알리는 음성 메시지는 동일한 내용의 음성 메시지를 다수의 사람들에게 반복적으로 발생하는 경우이고 은행에서 대출 이자 연체를 통지하는 음성 메시지의 경우는 고객의 이름과 연체이자 금액은 고객에 따라서 변하는 부분이고 시나리오의 나머지 부분은 모든 고객에게 동일한 시나리오가 적용될 것이다. 두 번째 요

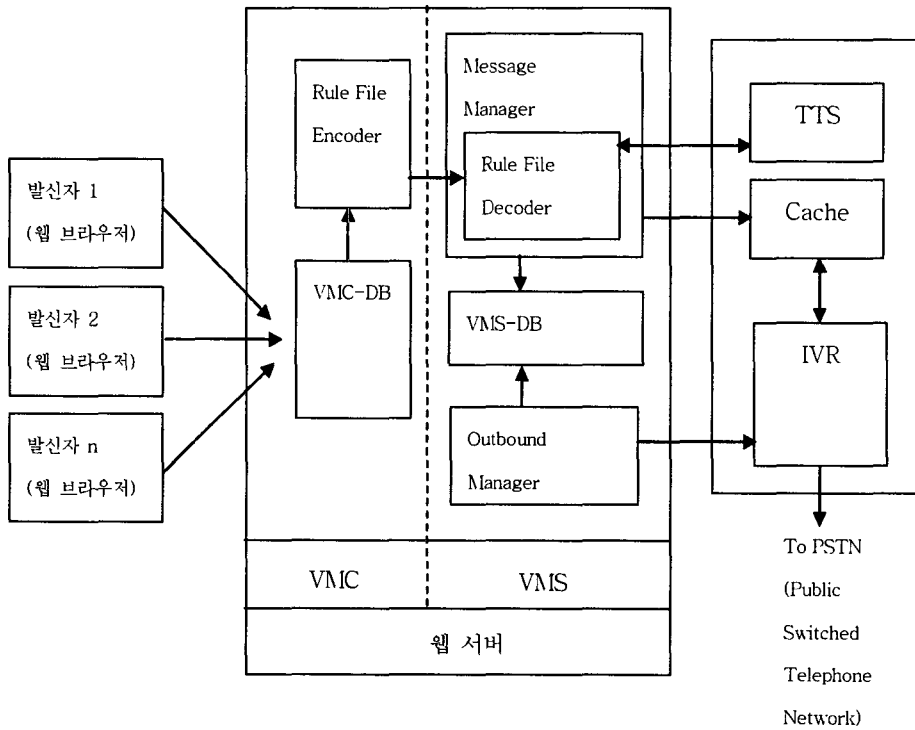
구 사항은 기존의 VoiceXML 기반의 IVR이 동일한 시나리오를 반복하는 경우에 VoiceXML 인터프리터는 한 번 실행한 시나리오에 관한 해독 정보를 저장해 두지 않고 그 시나리오가 요구될 때마다 매번 반복 작업을 수행하기 때문에 음성 메시지 서버에 많은 부담을 주게 된다. 이러한 문제를 해결하기 위하여 제안하는 기술은 서버에 부담을 주지 않고 메시지가 시스템에 발송되기에 적합한 최적의 상태가 되도록 변환되어 저장되어야 한다.

음성 메시지 시스템에서 응답속도를 지연시키는 요인은 IVR이 인터넷을 통하여 웹 서버에서 VoiceXML 문서, 오디오 파일, 문법, 스크립트, 데이터를 가지고 오거나 IVR이 여러 개의 Voice XML 문서 사이에서 이동(Transition)하는 경우, 또는 서브 다이얼로그의 호출 등과 같은 여러 가지 요인이 있을 수 있으나 본 논문에서 고려하는 시스템은 위의 두 번째 요구 사항에서 정의한 바와 같이 대량의 음성 메시지를 발송하는 시스템으로서 Voice XML 문서 사이의 이동이 적고 음성 메시지 수신자가 DTMF 입력을 사용하는 것으로 가정하였다. 이 같은 가정을 통하여 음성 인식 엔진이나 다른 요인들이 응답속도 지연에 미치는 영향이 작아서 응답속도 지연이 주로 TTS 엔진이나 서버 사이트 스크립트의 수행에서 오는 것으로 설정하였다.

본 논문에서는 서버 사이트 스크립트 사용으로 인한 웹 서버의 부담을 줄이고 TTS 엔진이 텍스트를 오디오 파일로 변환하거나 텍스트와 오디오를 합성하는 작업을 전처리 하여 IVR의 캐시 서버에 저장하여 응답지연을 제거하고 IVR의 성능을 향상시켜서 대량의 발신 호(Massive Outbound Call)를 지원할 수 있는 새로운 시스템 구현방법을 제안한다.

3.2. 제안된 시스템의 구성

제안하는 MOCASS의 구성은 다음과 같다. 제안하는 시스템에서는 웹 서버가 클라이언트 모듈(VMC:Voice Message Client)과 서버 모듈(VMS:Voice Message Server)로 나누어서 구성되어 있다. 앞에서 웹 서버와 IVR이 서버와 클라이언트의 관계로 구성되어 있는 것과 구별하기 위하여 웹 서버 내에서 음성 메시지 발신자의 요청을 받아서 Rule File을 생성하는 클라이언트 모듈은 VMC (Voice Message Client)라는 이름을 사용하였다. 클라이언트 모듈은 음성 메시지 발송자가 입력한 정보를 바탕으로 하여 음성 메시지 발송자가 발송을 원하는 메시지를 Rule File의 형태로 변환하여 서버 모듈로 전달하는 역할을 한다. Rule File은 Voice XML과 본 논문에서 새롭게 만든 태그가 혼합된(embedded) 형태로 되어 있다. <그림 3>의 새롭게 제안하는 시스템은 음성 메시지 발송자가 입력한 내용이 웹 서버 내부의 VMC를 통하여 VMS에 연결되는 점에서 <그림 2>와 다르다.



<그림 3> 제안하는 MOCASS의 구성도

서버 모듈의 Rule File Decoder는 클라이언트가 전송한 Rule File을 변환하여 VoiceXML 문서로 만들게 되는 데 이 때 만들어진 VoiceXML 문서는 서버에서 수행해야 하는 서버 사이드 스크립트를 포함하지 않고 텍스트의 TTS 변환을 필요로 하지 않는 즉시 실행이 가능한 순수한 VoiceXML 문서이다. 서버 모듈은 이러한 순수한 VoiceXML 시나리오를 IVR에게 전달하고 IVR은 다수의 수신자에게 음성 메시지를 전달한다. 다음에는 각 모듈 안의 구성 요소들의 기능을 설명한다. 먼저 VMC 내부의 구성 요소들의 기능을 설명한다.

3.2.1. 클라이언트 모듈: VMC-DB (Voice Message Client Database)

발신자가 보내려는 음성 메시지 발송에 필요한 기본적인 정보들이 저장된다. VMC 사용자가 MOCASS 사업자가 운영하는 웹 사이트를 방문하여 원하는 템플레이트에 따라서 발송하고자 하는 메시지 작성을 완성하면 VMC는 메시지 발송에 필요한 기본적인 정보들을 VMC 내부의 VMC-DB에 저장한다. 여기서 템플레이트

는 MOCASS 사업자가 제공하는 음성 메시지 발신 유형을 말하는 것으로서, 예를 들면 미수금 회수, 생일 파티 초대, 설문조사 등과 같은 미리 정해진 시나리오의 틀을 말한다. VMC- DB에 저장되는 정보의 몇 가지 예는 다음과 같다.

1. 발신자
2. 발송 내용
3. 발송에 관한 정보
 - * 발송 시간
 - * 주기 정보 (매일, 또는 일주일 간격)
 - * 발송에 실패했을 때의 대응 방안 등
4. 수신자.
5. 부가 서비스 내용
 - * 음성 메시지로만 발송되는지 또는 단문메시지(SMS)와 병행하여 발송되는지 여부 등

3.2.2. 클라이언트 모듈: Rule File Encoder

Rule File Encoder는 VMC-DB에 있는 기초적인 데이터들과 발신자가 지정한 템플레이트를 기초로 하여 Rule File을 생성하다. Rule File은 나중에 VMS가 순수한 VoiceXML 문서를 생성할 때 사용된다. Rule File안에서는 표준 VoiceXML 과 Rule File 전용 태그가 혼합되어 사용되며 아래는 Rule File 전용 태그의 몇 개의 예제를 보여준다.

예제1) `var variable name = Text2Audio` (“텍스트의 내용”)

위의 태그는 텍스트의 내용을 TTS를 통하여 오디오 파일 (대부분의 경우에 웨이브 파일)로 변환하여 변수(variable name)에 할당한다.

예제2) `var variable name 2 = AudioMix` (variable name 1, FileName)

위의 태그는 변수 이름(variable name 1)에 저장된 오디오 파일과 FileName (일반적으로 MIME; Multi-Purpose Internet Mail Extensions 형식)에 저장된 첨부 파일을 합성(Mix)하여 새로운 변수(variable name 2)에 저장한다.

예제3) `<audio src="<%=Response.Write(variable name);%>">`

VoiceXML의 audio 태그를 생성하는 태그로서 변수에 저장된 오디오 파일이 있는 URL (Uniform Resource Locator)의 위치를 나타내는 오디오 태그를 생성한다.

다음에는 VMS 안의 구성 요소들의 기능을 설명한다.

3.2.3. 서버 모듈: Rule File Decoder

Rule File Decoder는 VMC에게서 받은 Rule File 안에 정의된 태그를 해석하여 다음과 같은 기능을 수행한다.

1. 텍스트 부분을 음성 합성 엔진(TTS)으로 넘겨주고 음성 합성 엔진이 텍스트를 오디오 파일로 변환한 결과를 반환 받는다. (“Text2Audio” 태그)
2. AudioMix (variable name, FileName) 태그를 해석하여 음성과 배경 음악을 합성한다. 위의 1)에서 TTS가 변환한 결과가 저장된 변수(variable names)와 FileName이 지정하는 파일의 음악을 합성한다. FileName은 AudioMix에 참여하는 음악 파일의 위치를 지정하며 MOCASS가 제공하는 배경 하는 음악이 될 수도 있고 음성 메시지 발송자가 첨부한 배경음악이 될 수 있다. 합성에 필요한 배경 음악이 MOCASS 사업자가 제공하는 배경음악인 경우에는 음악을 다운로드 하여 합성을 한다. 만약에 합성에 필요한 배경 음악이 발신자가 첨부한 음악이면 (MIME 형식) 음악을 웨이브 파일로 변환하여 음성과 합성을 한다.
3. 위의 1), 2)에서 생성한 음성이나 음성과 배경 음악이 합성된 오디오 파일이 캐시 서버 내부에 저장되어 있는 위치(URL)를 나타내는 Voice XML의 오디오 소스 태그를 생성한다. (“audio src” 태그)

3.2.4. 서버 모듈: MM(Message Manager)

4. MM (Message Manager)는 위의 결과를 이용하여 Rule File로부터 순수한 VoiceXML 문서를 생성한다.
5. 음성 메시지 발신자가 다수의 수신자에게 음성 메시지를 발송하는 경우에는 Message Manager는 Rule File Decoder가 완성한 위의 결과를 다른 수신자의 경우에도 반복하여 적용한다. 예를 들어 생일 초대 음성 메시지를 10명에게 발송하는 경우에는, 한 개의 Rule File로부터 10개의 Voice XML 문서가 10개 생성이 된다. 이 때에 수신자의 이름과 발송자가 지정한 가변 부분만 각각 다르며 메시지의 내용은 동일한 VoiceXML 문서가 10개 생성된다.
6. MM은 위에서 생성한 Voice XML 문서를 IVR (또는 캐시 서버)에 저장하고

VoiceXML 문서가 저장된 주소, 발송 예정 시간, 발송과 관련된 정보를 VMS-DB에 등록을 한다.

7. 결과적으로 MM은 IVR이 실행하는 VoiceXML 문서의 위치를 VMS-DB에 등록하고 IVR이 실행하는 VoiceXML 문서 자체는 IVR 내부에 저장되며 VoiceXML 문서에 실행에 필요한 오디오 파일은 캐시 서버 내부에 저장된다. 이 때에 IVR과 캐시 서버는 물리적으로 하나의 서버 내부에 존재할 수도 있고 서로 별개의 서버로 분리된 상태에서 LAN (Local Area Network)이나 인터넷을 통하여 연결될 수도 있다.

3.2.5. 서버 모듈: VMS-DB

VMS-DB는 IVR이 수행하여야 할 VoiceXML 문서 (즉, 발송하여야 할 음성 메시지)가 저장된 주소 (URL)와 발송예정시간 등에 관한 정보를 저장한다.

3.2.6. 서버 모듈: OM (Outbound Manager)

OM은 VMS-DB를 주기적으로 조회하여 VoiceXML 문서의 발송예정시간을 확인하고 Voice XML 문서가 예정된 시간에 발송되도록 VoiceXML 문서의 위치를 IVR에게 통보한다.

3.3. 구현 예제

본 장에서는 제안하는 시스템에서 VoiceXML 문서 (음성 메시지)가 생성되는 예제를 보여준다. 아래의 예제는 음성 메시지 발송자가 고객에게 “감사를 나타내는 음성 메시지” 템플레이트를 선택하고 배경 음악과 함께 발송하는 예제를 보여주는 것이다. 음성 메시지 발송자가 MOCASS 사업자가 운영하는 홈페이지에 접속하여 다음과 같은 정보를 입력하고 이러한 내용은 VMC-DB에 저장이 된다.

- * 음성 메시지 발송자: 주식회사 갑을
- * 음성 메시지 수신자 1 이름: 김철수
- * 음성 메시지 수신자 1 전화 번호: 011-123-4567
- * 축하 메시지 수신자 1: 김철수

- * 음성 메시지 수신자 2: 이영희
- * 음성 메시지 수신자 2 전화번호: 016-234-5678

* 축하 메시지 수신자 2: 박수동

* 인사말 메시지: 안녕하세요? 주식회사 갑을 입니다.

* 본문 메시지: 갑을 전 직원은 “+ [축하 메시지 수신자]+”고객 님의“+[축하 메시지 수신자 나이]+”번째 생일을 진심으로 축하드립니다. (음성 메시지 발송자가 배경 음악을 지정)

위의 내용을 바탕으로 Rule File Encoder가 생성한 Rule File의 소스 코드는 다음과 같다.

1. <?xml version="1.0" encoding="euc-kr"?>
2. ...생략
3. <var name="ReserveCode" expr="88889999"/>
4. <var name="RealJobCode" expr="1"/>
5. <rule name='vmc_vxml'>
6. <![CDATA[
7. <?xml version="1.0" encoding="euc-kr"?>
8. <vxml version="2.0">
9. <form>
10. ... 생략
11. <field name="N">
12. ... 생략
13. <prompt>
14. <%= var temp1_wav=Text2Audio ("안녕하십니까? 주식회사 갑을 입니다.");%>
15. <audio src="<%=Response.Write(temp1_wav);%>"/>
16. <%= var temp2_wav=Text2Audio ("갑을 전 직원은" + QueryRecord ("info.info1") + "고객 님의" + QueryRecord("info.info2") + "번째 생일을 진심으로 축하드립니다.");%>
- 17.... 중간 생략
18. <%=var temp3_wav=AudioMix(temp2_wav, BgmFile);%>
19. <audio src="<%=Response.Write(temp3_wav);%>"/>
20. <audio src="http://URL1/cache/reprompt.wav"/>
21. </prompt>
22. </field>
23. ...No Input과 No Match는 생략

24. </vxml>
25.]]>
26. </rule>
27. <translate rule='vmc-vxml'>
28. <record>
29. <field name='info.ReserveCode'>88889999</field>
30. <field name='info.RealJobCode'>1</field>
31. <field name='info.RecipientName'>김철수</field>
32. <field name='info.Recipient'>011-123-4567</field>
33. <field name='info.info1'>김철수</field>
34. <field name='info.info2'>22</field>
35. </record>
36. </translate>
37. ... 생략
38. <translate rule='vmc-vxml'>
39. <record>
40. <field name='info.ReserveCode'>88889999</field>
41. <field name='info.RealJobCode'>2</field>
42. <field name='info.RecipientName'>이영희</field>
43. <field name='info.Recipient'>016-234-5678</field>
44. <field name='info.info1'>박수동</field>
45. <field name='info.info2'>30</field>
46. </record>
47. </translate>
48. ... 생략
49. content-disposition: attachment; file="BgmFile"
50. ... 첨부 파일 내용 생략

위의 Rule File은 Rule File Encoder에 의해서 VoiceXML 문서로 쉽게 변환되도록 하기 위하여 가능하면 VoiceXML 문서에 가깝도록 정의되었다.

위에서 라인 3의 ReserveCode는 IVR이 수행하는 작업 코드를 나타내며 라인 4의 RealjobCode는 발신자가 여러 명의 수신자에게 음성 메시지를 발송하는 경우에 수신자를 구분하기 위한 변수이다. ReserveCode와 RealjobCode는 IVR이 음성 메시지 발송 결과를 OM (Outbound Manager)에게 보고할 때 사용된다.

라인 14와 라인 19 사이에 < > 표시 내부의 "%="과 ";%"는 Rule File Decoder가 실행하는 부분을 표시한 제한자(delimiter)로 사용되었다. 라인 16의 "+" 문자는

텍스트와 가변하는 데이터가 연결(Concatenation) 되어서 오디오 파일로 변환되어 변수에 저장되는 것을 나타낸다. 또한 라인 16의 QueryRecord는 Rule File 밑에 있는 Translate 부분에 있는 Field Name을 정의하는 기능을 수행한다. 예를 들어서, ReserveCode가 88889999이고 RealJobCode가 “1”인 경우에는 QueryRecord (“info.info1”)는 “김철수”라는 결과를 받게 되고 ReserveCode가 88889999이고 RealJobCode가 “2”인 경우에는 QueryRecord (“info.info1”)는 “박수동”이라는 결과를 받게 된다. 라인 20에서 reprompt.wav 는 IVR이 음성메시지 끝부분에 추가한 내용으로서 “음성 메시지를 다시 듣기 원하시면 1번을 눌러주십시오”라는 오디오 파일을 나타낸다. 라인 27과 라인 36 사이의 Translate 태그 내용은 첫 번째 수신자에 관한 정보를 나타내고 라인 38과 라인 47 사이의 Translate 태그의 내용은 두 번째 수신자에 관한 정보를 나타낸다. 또한 Rule File의 마지막의 라인 49는 사용자가 첨부한 배경음악이 MIME 형식으로 첨부되어 있는 것을 나타낸다.

MM과 Rule File Decoder가 Rule File을 사용하여 생성한 VoiceXML 문서의 소스 코드는 다음과 같다.

1. <?xml version="1.0" encoding="euc-kr"?>
2. <vxml version="2.0">
3. <form>
4. <var name="ReserveCode" expr="88889999"/>
5. <var name="RealJobCode" expr="1"/>
6. <field name="N">
7. ... 생략
8. <prompt>
9. <audio src="http://URL2/cache/vmc1234.wav"/>
10. <audio src="http://URL3/cache/vmc2345.wav"/>
11. <audio src="http://URL1/cache/reprompt.wav"/>
12. No Input과 No Match는 생략
13. </field>
14. </vxml>

위 소스 코드의 라인 9의 URL2 밑의 vmc1234.wav는 “안녕하십니까? 주식회사 갑을 입니다.”를 들려주는 오디오 파일이며 라인 10의 URL3 밑의 vmc2345.wav는 “갑을 전 직원은 김철수 고객 님의 22번째 생일을 진심으로 축하드립니다.”를 들려주는 오디오 파일이다. URL2나 URL3은 캐시 서버의 IP 주소를 나타낸다. 또한 라인 11의 reprompt.wav는 “다시 듣고 싶으시면 1번을 눌러 주십시오.”라는 음성을 들려주는 오디오 파일을 나타낸다.

이미 설명한 바와 같이 MM은 음성메시지 수신자 “이영희”에게 발송될 VoiceXML 문서도 위와 같은 방법으로 생성하며 이 문서는 위와 같이 세 개의 각각 다른 URL에 “안녕하십니까? 주식회사 갑을 입니다.”, “갑을 전 직원은 박수동 고객 님의 30번째 생일을 진심으로 축하드립니다.”, “다시 듣고 싶으시면 1번을 눌러 주십시오.”라는 음성 파일이 저장된 위치를 지정하게 된다.

위에서 보는 바와 같이 Rule File Decoder가 생성한 VoiceXML 문서의 특징은 다음과 같다.

1. VoiceXML 문서가 어떠한 형태의 서버 사이드 스크립트를 포함하지 않는다.
2. VoiceXML 문서는 텍스트나 텍스트와 오디오 파일의 합성이 전처리를 거쳐 완료되어 실행 직전의 오디오 파일로 저장된 위치를 지정하므로 IVR이 TTS를 구동하지 않고 실시간으로 즉시 VoiceXML 문서를 실행하여 응답 지연의 문제를 해결하였다.

4. 시범 운영 사례

본 논문에서 제안하는 시스템은 현재 음성메시지 응용 프로그램 임대 사업을 하고 있는 국내의 K사에서 운영하는 음성 메시지 발송 서비스에 시범적으로 적용되었다. K사는 새로운 음성 메시지 발송 시스템을 적용하기 이전에 <그림 2>에서 설명한 기존의 음성 메시지 발송 시스템을 Windows NT와 ASP 인터페이스를 갖는 IIS 기반의 웹 서버와 음성메시지 IVR을 사용하여 운영하였고 IVR은 90개의 전화 회선에 연결하여 운영하였다.

K사는 많은 수의 음성 메시지 발송자가 웹 사이트에 접속하여 음성 메시지 발송을 동시에 요청하여 90개의 회선이 모두 가동되었을 때에 일부의 음성 메시지 수신자는 IVR의 발신 호(Outbound Call)를 받은 후에 음성 메시지 재생이 2~3초 정도 지연되는 문제점을 겪었다. 이 같은 현상의 원인으로서는 앞에서 지적한 바와 같이 웹 서버의 서버 사이드 스크립트 수행을 위한 DB 접속이나 연속적인 TTS 구동에 따른 응답 지연에 기인하는 것으로 추측된다.

K사는 국내의 A 택배회사에 보다 나은 음성 메시지 발송 서비스를 제공하기 위하여 본 논문에서 새롭게 제안한 시스템을 구성하여 응답 지연의 문제점을 해결하였다. 운영 환경은 웹 서버는 Windows 2000 서버와 IIS 4.0을 사용하였으며 (주)미디어포드에서 C, C++로 개발한 Rule File Ender/Decoder가 설치되었으며 VMC-DB/VMS-DB에는 MS-SQL이 사용되었다.

그리고 IVR에는 Windows 2000 서버, 2개의 Dialogic 음성처리 보드 D/600JCT-2E1이 설치되었고 (주)미디어포드에서 C, C++로 개발한 Voice XML 인

터프리티어가 사용되었다. A 택배 회사는 최대 연속 1000 통화 정도의 음성 메시지를 발송하였으며 이 경우에 음성 메시지 수신자가 감지할 정도의 응답지연은 나타나지 않았다.

5. 결론 및 향후 과제

본 논문에서는 대량의 음성 메시지를 발송하는 음성 메시지 응용 프로그램 임대 사업자에게 적합한 음성 메시지 발송 시스템을 제안하였다. 본고에서 제안한 MOCASS는 웹 서버가 ASP와 같은 서버사이드 스크립트를 사용하는 대신에 Rule file을 사용하여 서버의 부하를 VMS(서버)와 VMC(클라이언트) 사이에 분산시켰고 또한 대량의 음성 메시지가 TTS에 동시에 접근하는 부담을 줄이기 위하여 VoiceXML 문서에서 지정하는 텍스트, 오디오 파일, 텍스트와 오디오의 합성 파일을 음성 메시지 발송 전에 전처리 하여 미리 메시지 발송에 필요한 형태로 변환하여 IVR이 전용으로 접근할 수 있는 캐시 서버에 저장하여 응답지연의 문제점을 해결하였다.

제안하는 시스템을 채용한 K사의 음성메시지 서비스는 현재 가동 중인 사이트로서 기존 시스템의 응답지연의 정도나 정확한 원인 그리고 새로운 음성 메시지 시스템을 채용한 후의 성능 개선의 정도를 측정할 수 있는 환경이 제공되지 않았다. 향후 연구에서는 정해진 하드웨어, 소프트웨어 환경에서 포트 밀도(port density)와 TTS 부하를 모의 실험하는 포트 밀도 시뮬레이터(port density simulator)를 기존의 시스템과 제안하는 시스템에 연결하여 응답 지연을 정도를 모의실험하고 실제 운영 결과와 비교하여 제안하는 시스템이 효율적인 MOCASS 구축 방안을 제시하는지 체계적으로 평가하는 것이 필요한 것으로 보인다.

또한 웹 서버가 내부적으로 VMC와 VMS로 분리되어 있는 현재의 구성은 발신 호의 증가에 따라 Rule File Encoder의 부하가 증가하여 웹 서버에 부담을 증가시킬 수 있으므로 음성 메시지 발송자에게 VMC (Rule File Encoder)의 기능을 내장한 음성 메시지 발송 전용 클라이언트 모듈을 배포하고 기존의 웹 서버의 부하를 음성 메시지 발송자(VMC)와 웹 서버(VMS)로 분리하여 MOCASS 전체의 구성을 음성 메시지 발송자 (VMC) - 웹 서버 (VMS) - IVR의 세 계층(3-tier architecture)으로 분산시켜서 더욱 안정적인 시스템을 구축할 수 있는 연구가 필요한 것으로 보인다.

참 고 문 헌

- [1] Voice XML(Voice eXtensible Markup Language) Version 2.0,
http://www.w3.org/Voice/2003/ir/Voice_XML20-ir.html,
 W3C Candidate recommendation, April 7, 2003.
- [2] 홍기형, “음성 정보 기술 응용 개발 표준화”, *전자공학회지*, 30권, 7호, pp.45-52, 2003.
- [3] <http://studio.tellme.com/vxml2/ovw/fetchaudio-p.html>
- [4] <http://studio.tellme.com/vxml2/ovw/perf/caching-p.html>
- [5] 박섭형, 음성 웹 애플리케이션 구축을 위한 *Voice XML*, 한빛미디어, 2001.
- [6] http://studio.tellme.com/dom/howto/using_data-p.html
- [7] <http://vxml.widerthan.com/docu/expand04.php>
- [8] <http://www.cs.columbia.edu/~hgs/teaching/ais/1998/projects/ebp/report.html#TTS>

접수일자 : 2004년 2월 22일

게재결정 : 2004년 3월 9일

▶ 김정곤(Jeonggon Kim)

주소: 경기도 군포시 당정동 604-5 한세대학교

소속: 한세대학교 IT 학부

전화: 031) 450-5174

E-mail: jeongkim@hansei.ac.kr