



저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

홍 기 형 교수지도
석사학위 청구논문

명령형 음성 사용자 인터페이스의
사용성 평가를 위한 도구 설계 및 구현

2011

성신여자대학교 대학원
컴퓨터학과
이 명 지

명령형 음성 사용자 인터페이스의 사용성 평가를 위한 도구 설계 및 구현

홍 기 형 교수지도

이 논문을 석사학위논문으로 제출함

2010년 11월

성신여자대학교 대학원
컴퓨터학과
이 명 지

인 준 서

이명지의 석사학위 논문으로 인준함.

심사위원 _____인

심사위원 _____인

심사위원 _____인

성신여자대학교 대학원

논문 개요

음성 사용자 인터페이스(Voice User Interface) 시스템의 상용화 사례가 증가함에 따라 음성 사용자 인터페이스의 사용성을 향상시키는 것이 중요시 되고 있다. 따라서 음성 사용자 인터페이스 설계 시 사용성에 초점을 맞춘 사용자 중심 설계 방법을 적용할 필요가 있다.

음성 사용자 인터페이스를 구성하는 요소에는 프롬프트, 인식 문법, 대화 로직이 있다. 본 논문에서는 음성 사용자 인터페이스 설계 시 구성 요소 중 프롬프트와 인식 문법의 사용성을 향상시키기 위해 위자드 오브 오즈 방식의 초기 사용성 평가 도구를 구현하였고, 이 도구에 사용될 다이얼로그 시뮬레이션 전용 문서인 VUIDML(Voice User Interface Design Markup Language)을 제안하였다. VUIDML은 VoiceXML을 수정하고 재구성한 것으로 명령형 음성 인터페이스의 사용성 평가 시나리오를 위한 XML 기반 언어이다. VUIDML 문서와 도구를 통해 실사용자가 기억하기 쉬운 음성 명령을 수집할 수 있고, 프롬프트 선호도를 파악할 수 있다.

목 차

논문 개요

I. 서론	1
II. 관련 연구	3
1. 사용성	3
1) 사용성의 정의	3
2) 사용성 향상 원리	4
2. 사용자 중심 설계	5
3. 사용성 평가	7
1) 사용성 평가의 정의 및 특성	7
2) 사용성 평가 절차	7
3) 위자드 오브 오즈	8
4. 음성 사용자 인터페이스	9
5. 기존 사용성 평가 도구	10
1) 동기·비동기 분류	10
(1) 동기식 사용성 평가 도구	10
(2) 비동기식 사용성 평가 도구	10
2) 사용 목적에 따른 분류	10
(1) 설문 제작 및 배포	10

(2) 전문가에 의한 디자인 사용성 평가	11
(3) 인터랙션 사용성 평가	11
(4) 음성 사용자 인터페이스 사용성 평가	11
III. 명령형 음성 사용자 인터페이스의 사용성 평가를 위한 도구 설계 ...	12
1. 음성 인터페이스 설계용 초기 사용성 평가 도구 설계	12
1) 음성 인터페이스의 사용성 향상 요소	12
2) 명령형 음성 인터페이스	13
3) 도구 설계 및 구현 범위	16
4) 기능 요구 사항	18
2. Voice User Interface Design Markup Language	21
1) 기능 및 특징	21
2) 구성 요소	22
(1) <vuidml> 엘리먼트	23
(2) <meta> 엘리먼트	24
(3) <form> 엘리먼트	25
(4) <block> 엘리먼트	26
(5) <field> 엘리먼트	28
(6) <prompt> 엘리먼트	29
(7) <noinput>, <nomatch>, <error> 엘리먼트	31
(8) <match> 엘리먼트	32

(9) <failed> 엘리먼트	34
(10) <candidate> 엘리먼트	35
(11) <audio> 엘리먼트	36
(12) <var>, <value> 엘리먼트	38
3. VUIDML 문서 활용 예	40
4. VUIDML을 이용한 사용성 평가 과정	43
IV. 명령형 음성 사용자 인터페이스의 사용성 평가를 위한 도구 구현 ...	45
1. 구현 환경	45
2. 사용 환경	46
3. 시스템 구성	47
4. VUIDML 파싱 절차	52
1) Document 인스턴스 생성 및 저장	52
2) <var> 엘리먼트 추출	52
3) <var> 엘리먼트 값 위자드 지정	52
4) 다이얼로그 컨텍스트 객체 생성	53
5) Prompt 객체의 TTS 텍스트 저장	54
6. 위자드-서브젝트 통신	56
1) MessageTransfer 관리자	56
2) FileTransfer 관리자	58
7. 시스템 수행 과정	59

1) 프롬프트 출력	60
2) 발화 녹음	63
3) 선호도 조사	65
V. 결론	68
참고문헌 및 사이트	
ABSTRACT	

그림 목차

[그림 2-1] 사용자 중심 설계 모델	6
[그림 3-1] 다이얼로그 모델	14
[그림 3-2] 음성 인터페이스의 사용자 중심 설계 과정	16
[그림 3-3] 도구 구현 범위	17
[그림 3-4] VUIDML 구조	22
[그림 3-5] <vuidml> 엘리먼트 작성 예	23
[그림 3-6] <meta> 엘리먼트	25
[그림 3-7] <form> 엘리먼트 작성 예	26
[그림 3-8] <block> 엘리먼트 작성 예	27
[그림 3-9] <block> 엘리먼트 작성 예	29
[그림 3-10] <prompt> 엘리먼트 작성 예	30
[그림 3-11] <match> 엘리먼트 작성 예	33
[그림 3-12] <failed> 엘리먼트 작성 예	35
[그림 3-13] <failed> 엘리먼트 작성 예	37
[그림 3-14] <var> 와 <value> 엘리먼트 사용 예	39
[그림 3-15] VUIDML 문서 작성 예1	40
[그림 3-16] VUIDML 문서 작성 예2	42
[그림 3-17] VUIDML 시뮬레이션 과정	43
[그림 4-1] 도구 개념도	47

[그림 4-2] 위자드 시스템	48
[그림 4-3] 서브젝트 시스템	50
[그림 4-4] VarObj 오브젝트	52
[그림 4-5] 다이얼로그 컨텍스트 객체 구조	53
[그림 4-6] 프롬프트 출력	60
[그림 4-7] VUIDML 문서 선택 다이얼로그	61
[그림 4-8] 변수 설정 다이얼로그	61
[그림 4-9] 발화 녹음	63
[그림 4-10] 선호도 조사	65
[그림 4-11] 프롬프트 선호도 조사 화면	66
[그림 4-12] 선호 프롬프트 저장 구조	67

표 목차

[표 3-1] <vuidml> 엘리먼트	23
[표 3-2] <meta> 엘리먼트	24
[표 3-3] <form> 엘리먼트	25
[표 3-4] <block> 엘리먼트	26
[표 3-5] <field> 엘리먼트	28
[표 3-6] <prompt> 엘리먼트	30
[표 3-7] <match> 엘리먼트	32
[표 3-8] <failed> 엘리먼트	34
[표 3-9] <candidate> 엘리먼트	35
[표 3-10] <audio> 엘리먼트	36
[표 3-11] <var> 엘리먼트	38
[표 3-12] <value> 엘리먼트	38
[표 4-1] 구현 환경	45
[표 4-2] 사용 권장 사양	46
[표 4-3] 워자드 시스템 송/수신 메시지	56
[표 4-4] 서브젝트 시스템 송/수신 메시지	57
[표 4-5] 시스템 상태	59

I. 서론

스마트 폰과 같은 모바일 기기나 내비게이션과 같은 전자 제품, 가전 제품, IPTV 서비스 등 다양한 분야에 음성 인식 기반의 음성 사용자 인터페이스가 적용되어 상용화되는 사례가 증가하고 있다. 음성 사용자 인터페이스를 적용한 제품에 대한 사용자의 수요 또한 증가하고 있다[1][2][3]. 이에 따라서 음성 사용자 인터페이스를 시스템에 적용했을 때 제품의 사용성이 어떻게 증가하는지 또는 어떻게 해야 사용성을 향상시킬 수 있는지에 대한 문제가 논의되고 있다[4][5].

음성 사용자 인터페이스 설계 단계에서 고려되는 요소는 프롬프트, 인식 문법, 대화 로직이 있다. 음성 사용자 인터페이스의 사용성 향상을 위해서는 이 세 가지 구성 요소 설계 시 사용성 향상에 중점을 두고 작업을 진행해야 한다. 음성 사용자 인터페이스의 구성 요소의 사용성을 향상시키기 위해서는 반복적인 설계가 요구된다[6].

사용자 중심 설계 방법은 반복적인 설계를 원칙으로 한다[7]. 이를 음성 사용자 인터페이스의 구성 요소 설계에 적용하는 경우 사용성 평가를 통해 프롬프트의 선호도 조사, 명령어 수집, 대화 흐름 유형 수집과 그 결과를 활용한 구성 요소의 재설계가 가능하다. 이 과정에서는 프롬프트, 인식 문법, 대화 로직만 설계되고 최종 사용자가 사용하게 될 시스템이 완전히 구현된 상태가 아니므로, 시뮬레이션을 통한 초기 사용성 평가가 실시되는 것이 적합하다. 초기 사용성 평가는 위자드 오브 오즈 방식으로 진행되어 평가에 참가하는 실제 사용자가 평가 대상인 시스템이 제대로 작동된다고 인지하도록 시뮬레이션을 할 수 있게 준비되어야 한다.

본 논문에서는 명령형 음성 인터페이스 설계 시 실시할 초기 사용성 평가

를 자동화하기 위하여 위자드 오브 오즈(Wizard Of Oz)방식의 시뮬레이션 도구를 설계하고 구현하였다. 초기 사용성 평가에서 실시될 시뮬레이션 대상은 음성 사용자 인터페이스의 세 가지 구성 요소 중 프롬프트, 인식 문법으로 제한하였다. 또한 VoiceXML[8][9][10] 구성 요소를 수정하고 재구성하여, 명령형 음성 인터페이스 시뮬레이션의 사용성 평가용 시나리오 작성을 위한 VUIDML(Voice User Interface Design Markup Language)을 설계하여 제안하였다. VUIDML은 XML 기반 언어로서 다양한 대화 흐름과 후보 프롬프트들을 포함하도록 다이얼로그 시나리오를 작성하여, 실사용자가 사용하기 편한 음성 명령과 선호하는 프롬프트를 수집하는데 사용할 수 있다.

본 논문의 구성은 다음과 같다. 2 장에서는 관련 연구로서 사용성과 사용자 중심 설계, 사용성 평가 및 음성 사용자 인터페이스와 기존에 개발되어 있는 사용성 평가 도구 사례에 대하여 기술한다. 3 장에서는 명령형 음성 인터페이스의 사용성 평가 도구를 설계하기 위한 음성 사용자 인터페이스의 사용성 향상 요소와 명령형 음성 인터페이스의 다이얼로그 모델 및 도구의 기능 요구 사항에 대하여 기술하고, 사용성 평가에 사용하기 위해 제안한 VUIDML에 대해 설명한다. 4 장에서는 명령형 음성 인터페이스 설계를 위한 사용성 평가 도구를 구현한 내용에 대하여 기술한다. 마지막으로 5장에서는 결론을 맺는다.

II. 관련 연구

1. 사용성

1) 사용성의 정의

사용성은 제품의 기능성처럼 모든 제품의 특성 중 하나이다. 기능성은 제품이 무엇을 할 수 있는지를 가리킨다. 기능성을 시험한다는 것은 제품이 제품 명세서대로 작동하는지를 보는 것이다. 반면 사용성은 사용자가 제품을 어떻게 다루는가를 나타낸다.

넬슨은 사용성이란 사용자 인터페이스가 얼마나 사용하기 쉬운지를 평가하는 질적인 속성이라고 정의하였다[11]. 듀마와 래디쉬(Duma and Radish)는 사용성은 사용자가 제품을 사용하여 직무를 빠르고 쉽게 완수하는 것이라고 정의하였다[12].

넬슨은 사용성은 다음 다섯 가지 질적인 요소에 의해 정의된다고 하였다 [11][13].

- 학습용이성(Learnability): 사용자가 제품을 처음 접했을 때 얼마나 쉽게 기본적인 작업을 수행하는지를 나타낸다.
- 효율성(Efficiency): 사용자가 일단 제품에 대해 이해했다면, 얼마나 빠르게 작업을 수행하는지를 나타낸다.
- 기억용이성(Memorability): 사용자가 제품을 어느 기간 동안 사용하지 않은 후에 제품을 사용했을 때 얼마나 쉽게 다시 능숙하게 사용할 수 있게 되는지를 나타낸다.

- 오류(Error): 사용자가 오류를 얼마나 많이 접하는지, 이 오류들의 심각도는 어느 정도인지, 그리고 얼마나 쉽게 오류에서 빠져나오는지를 나타낸다.
- 만족(Satisfaction): 제품을 사용했을 때 얼마나 즐거워하는지를 나타낸다.

2) 사용성 향상 원리

굴드와 르위스(Gould and Lewis)는 사용성을 향상시키기 위한 일반적인 세 가지 원리를 다음과 같이 설명하였다[13][14].

- 처음부터 사용자와 작업에 초점을 둔다.
사용자 프로파일링, 작업 분석, 프로토타이핑, 사용자 검토회(walkthrough)와 같은 작업을 수반한다.
- 실증적인 척도와 방법을 적용한다.
설문 실시, 실험실 또는 필드 사용성 연구, 사용(Usage) 연구와 같은 작업과 기법들이 객관적이고 정량적인 성능과 만족도에 대한 자료를 수집하는 것을 가능하게 해준다.
- 반복적으로 설계한다.
사용자 인터페이스 관리 시스템을 사용하여 구현된 시스템은 실증적인 데이터에 대하여 인터페이스가 빠르고 쉽게 근본적으로 변화될 수 있도록 해준다. 이는 반복적인 시험과 재설계를 가능하게 한다.

사용성 향상의 원리는 사용자 중심 설계의 기본 원칙과도 일치한다.

2. 사용자 중심 설계

시스템의 설계 시 시스템이 가져야 하는 기능의 검증이나 성능보다는 사용성에 초점을 맞추어 설계하는 방법론을 사용자 중심 설계라고 한다.

일반적인 사용자 중심 설계에는 다음과 같은 원칙이 있다[7].

- 사용자와 사용자가 수행하는 작업에 먼저 초점을 둔다. 사용자에 대한 이해, 사용자가 수행할 작업에 대한 이해, 그리고 사용자의 작업 수행 환경에 대한 이해를 먼저 한다.
- 사용성 평가는 실 사용자를 대상으로 실험과 관찰을 통하여 시행한다.
- 사용성 평가를 시스템 개발 생명 주기에 걸쳐 반복 시행하고 그 결과를 반영하여 반복적으로 설계를 한다.

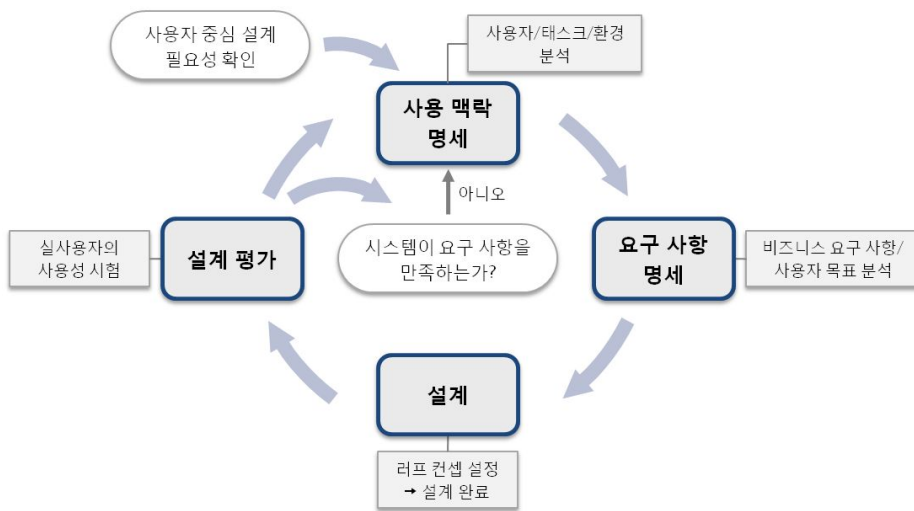
즉, 사용자 중심 설계를 위해서는 시스템의 설계 시 사용성 평가와 평가 결과를 반영해 재설계하는 단계가 반복적으로 수행되어야 함을 의미한다.

시스템의 설계 방법을 결정할 때, 설계자 또는 개발자가 사용자 중심 설계의 필요성을 인식하면 [그림 2-1]에서 보이는 것과 같이 4 단계의 과정을 거쳐 진행된다[15].

- 사용 맥락 명세: 사용자 중심 설계에서 시스템을 사용하게 될 실제 사용자에게 대해 분석하여 사용자 프로필을 작성하고, 사용자가 시스템을 무엇을 위해 이용할 것인지 태스크를 분석한다. 마지막으로 사용자가 시스템을 어떠한 상황과 조건에서 이용할지 사용 환경을 분석한다.
- 요구 사항 명세: 시스템이 성공적이기 위해서 반드시 충족시켜져야 하는 비즈니스 요구 사항 및 사용자가 시스템을 통해 달성하고자 하는 목표를 분석하여 요구 사항을 명세한다.
- 설계: 실제로 시스템을 설계하는 과정이다. 시스템의 대략적인 컨셉을

설정하는 단계에서 출발하여 상세하게 설계를 완료하는 작업을 거친다.

- 설계 평가: 4 단계 중 가장 중요한 부분으로 앞 단계에서 설계한 시스템을 실제로 평가한다. 시스템의 실제 사용자를 대상으로 사용성 평가를 실시하는 것이 이상적이다. 평가를 통해 수집된 결과를 분석하고 설계된 내용이 시스템의 요구 사항을 만족하는지를 검토한다. 만족하지 않는다면 첫 번째 단계인 ‘사용 맥락 명세’로 돌아가 반복적으로 설계를 진행한다.



[그림 2-1] 사용자 중심 설계 모델

3. 사용성 평가

1) 사용성 평가의 정의 및 특성

사용성 평가란 평가의 대상이 되는 제품의 사용성을 향상시키기 위해 제품의 실제 사용자를 대상으로 제품을 이용하여 특정 작업을 수행하도록 하여 사용성을 측정하는 시험이다. 즉, 사용성 평가는 사용자가 본인의 필요에 의해 제품의 해당 기능을 찾아서 사용하는 것을 보는 것이다[12].

사용성 평가의 기본적인 특성은 다음과 같다[12].

- 제품의 사용성을 향상시키는 것이 기본적인 목표이다. 각각의 테스트 특수 목표와 관심사항을 가지고 있다.
- 참가자는 실사용자를 대표한다.
- 참가자는 현실적인 과제를 수행한다.
- 사용성 평가자는 참가자의 말과 행동을 관찰하고 기록한다.
- 사용성 평가자는 데이터를 분석하고, 문제점을 진단하고 해결방안을 제시한다.

사용성 평가는 참가자가 최소한 2~3명은 되어야 개인차에 따른 행동을 걸러낼 수 있고, 사용성 평가를 통해 드러난 문제를 확증시켜 주기 위해서는 특정 문제를 겪은 참가자의 수, 잘못된 선택의 횟수, 과제를 마무리하는 데 걸린 시간과 같은 정량적인 기본 측정치가 필요하다. 이 측정치는 관찰, 데이터 기록 프로그램, 비디오테이프 분석을 통해 얻어낼 수 있다.

2) 사용성 평가 절차

대표적인 사용성 평가의 과정은 크게 세 부분으로 나누어진다. 첫 번째

단계에서 실제 사용 상황을 가정하여 미리 작성된 태스크 시나리오에 맞춰 기기 조작을 수행한다. 이 때 빈번하게 발생하는 태스크 수행 시간, 수행도, 특히 사항, 주요 오류 유형을 측정하고 기록한다. 두 번째 단계에서는 제품에 대한 사용자 인터페이스의 주요 구성 요소 별 소비자 만족도를 평가하여 해당 제품의 사용성을 높이기 위해서 우선적으로 개선 보완해야 할 점을 도출한다. 세 번째 단계에서는 주요 사용상 오류의 원인을 진단하고 이를 개선 및 보완하기 위한 아이디어를 구상하고 제시한다[16].

3) 위자드 오브 오즈

위자드 오브 오즈(Wizard Of Oz; WOZ) 기법은 아직 존재 안 하는 시스템을 사용성 평가하는 사용성 평가 기법이다. 이는 시스템 설계자가 시스템을 구현하지 않고 제품을 사용성 평가할 수 있도록 해 준다[17]. 위자드 오브 오즈 방식은 다음과 같이 동작된다.

- 사용자는 스크린과 상호작용한다.
- 사용자가 스크린을 통해 요청하면, 소프트웨어가 응답하는 것이 아니라 개발자(위자드)가 다른 스크린(일반적으로 다른 실험실 공간)에서 시스템을 시뮬레이션 하여 사용자와 상호작용한다.
- 위자드는 시스템 기능의 전부 또는 일부를 시뮬레이션 한다.
- 위자드 오브 오즈를 위한 시뮬레이션을 준비할 때, 이전에 구현된 시스템에 대한 경험이 위자드 ‘능력’의 실제적인 범위를 정하는데 도움이 된다.

위자드 오브 오즈는 프로토타입을 사용성 평가하고 시스템에 대한 사용자의 기대를 수집하는데 이상적이다. 개발자는 프로그래밍하는 비용 없이 사용성 평가 중에 인간의 간섭을 통해 누락된 기능을 제공함으로써, 제한된 기능을 구현한 프로토타입의 기능을 강화시킬 수 있다.

4. 음성 사용자 인터페이스

음성 사용자 인터페이스(Voice User Interface; VUI)란 사람과 음성 언어 어플리케이션이 서로 상호 작용하는 것으로 프롬프트(시스템 메시지), 문법, 대화 로직(call flow)으로 구성된다[18][19].

- **프롬프트:** 대화 중에 시스템이 사용자에게 들려주는 합성된 음성 또는 미리 녹음된 음성을 말한다.
- **인식 문법:** 각각의 프롬프트에 대한 사용자의 모든 가능한 응답을 정의한다. 시스템은 문법에 포함되어있는 단어, 문장, 구문들만 인식한다.
- **대화 로직:** 시스템에 의해 발생할 수 있는 모든 가능한 행위를 정의한다. 사용자의 발화에 대한 응답이나 데이터베이스에서 조회된 정보를 음성으로 출력해주는 등의 행위를 예로 들 수 있다.

음성 사용자 인터페이스의 특징 중 고유한 특징으로 다음의 두 가지가 두드러진다. 모달리티는 청각이고, 인터랙션은 음성 언어를 통해 이루어진다는 것이다. 음성은 인간이 사용하는 가장 자연스러운 의사소통 수단이지만, 이를 인터페이스로 하여 유용하고 신뢰성 있게 설계하기 위해서는 사용자 중심 설계를 통해 사용성을 향상시켜야 한다[20].

위에서 언급한 세 가지 음성 사용자 인터페이스 구성 요소를 설계하는 과정에서 사용자 중심 설계 방법을 적용하여 초기 사용성 평가를 실시해야 한다[21][22]. 이 때 초기 사용성 평가에서 사용자가 ‘완전히 개발된 시스템을 사용하고 있다.’라고 믿게 하기 위해서 위자드 오브 오즈 기법을 적용하여 사용성 평가를 실시하는 것이 유용하다[7].

5. 기존 사용성 평가 도구

사용성을 평가하기 위한 전형적인 방법은 피실험자가 현재 사용성 평가시 활용할 수 있는 도구의 사례들을 조사하였고 조사한 도구들의 유형을 진행 방식에 따라 2 가지로 분류하였고, 사용 목적에 따라 4 가지로 분류하였다.

1) 동기·비동기 분류[23][24]

(1) 동기식 사용성 평가 도구

1 대 1 방식으로 원격 사용성 평가 도구를 통해 사용성 평가를 진행하는 형태이다. 사용성 평가를 정해진 시나리오에 따라 실시 할 수 있어 평가 과정의 통제가 가능하고, 문제 발생 시 시험자가 평가 중간이나 종료 후에 해결이 가능하다. 또한 평가에 참가한 사용자를 직접 관찰하고 사용자의 반응과 의견을 기록할 수 있다. 그러나 사용자가 평가 대상을 사용해보는 실제 환경이 아니기 때문에 발견되지 않는 사용성 문제들이 존재할 수 있다.

(2) 비동기식 사용성 평가 도구

일반적으로 온라인 형태의 웹 페이지로 기능이 제공된다. 동시에 다수의 사용자가 실제 사용 환경에서 평가를 실시 할 수 있어 비용이 동기식에 비해 적게 들고 수집한 데이터의 양은 많다. 그러나 평가 과정의 사용자를 관찰할 수 없기 때문에 시험자와 사용자간의 상호작용이 발생하지 않는다. 또한 분석된 데이터의 질이 동기식에 비해 낮다.

2) 사용 목적에 따른 분류[25][26]

(1) 설문 제작 및 배포

온라인에서 마케팅 자료 수집 또는 연구 목적으로 설문을 제작하고 배포

하는 목적으로 사용 가능하다.

(2) 전문가에 의한 디자인 사용성 평가

전문가에 의해 웹 사이트의 디자인이나 인터페이스에 대한 의견을 빠르게 얻을 수 있다(Concept Feedback). 단, 실제 작업을 수행하여 평가하는 것이 아니라 웹 사이트의 그림만 보고 평가하기 때문에, 인터랙션에 대한 의견을 얻을 수 없다. 평가자 의견이 사용자의 의견을 반영한다고 보장하기 어렵다.

(3) 인터랙션 사용성 평가

웹 사이트의 최적화를 위해 사용자의 인터랙션 사용성 평가를 실시한다. 이때 평가 기준이 되는 방식은 웹 사이트 내의 그림이나 메뉴, 버튼들의 클릭 상태를 추적(Chalkmark, ClickHeat, ClickTale, Clixpy)하거나 두 개 이상의 다른 구성의 웹 페이지를 대상으로 하여 각각의 트래픽을 분석한 후 최종으로 한 개의 웹 페이지를 선택(Google Website Optimizer)한다.

(4) 음성 사용자 인터페이스 사용성 평가

음성 사용자 인터페이스의 사용성을 평가하기 위해 개발된 도구의 사례로 SUEDE[27]이 있다. 이는 다이얼로그에 사용되는 시나리오를 작성하고 실제 평가 과정에서 사용자의 발화를 녹음하는 기능을 제공한다. 단, 출력되는 프롬프트는 사전에 녹음된 오디오 파일로만 추가가 가능하다.

일반적인 사용성 평가를 지원하는 도구는 음성 사용자 인터페이스를 구성하는 프롬프트, 인식 문법을 대상으로 사용성을 평가하기 위한 오디오 출력 및 녹음 기능을 제공하지 않는다. 기존의 VUI 사용성 평가 도구는 설계 단계에서 프롬프트의 선호도를 조사하는 것이 불가능하고 음성 인식의 결과를 선택하여 텍스트를 오디오로 실시간으로 변환하여 시뮬레이션할 수 없다.

III. 명령형 음성 사용자 인터페이스의 사용성 평가를 위한 도구 설계

1. 음성 인터페이스 설계용 초기 사용성 평가 도구 설계

1) 음성 인터페이스의 사용성 향상 요소

음성 인터페이스 설계용 초기 사용성 평가 도구에서 구현되어야 할 기능을 수집하고 분석하기 위해서, 2 장에서 소개한 음성 사용자 인터페이스의 세 가지 구성 요소 설계 시 사용성 향상을 위해 고려해야 할 사항들은 다음과 같다.

프롬프트의 사용성을 향상시키기 위해서는 사용자로부터 선호하는 프롬프트의 유형을 수집해야 한다. 프롬프트는 실제 사용자가 이해할 수 있는 사용자 어휘들로 구성되어야 한다. 만약 사용자가 접하기 어려운 전문 용어나 시스템 설계자나 개발자들이 사용하는 단어를 포함하여 프롬프트를 구성하였을 경우, 사용자는 프롬프트를 듣고 이해하기 힘든 불편함을 겪게 된다. 이와 같은 문제의 발생을 막기 위해 사용성 평가를 통해 프롬프트에 대한 이해도를 조사해야 한다. 또한 여러 개의 대안이 있는 프롬프트들을 차례대로 사용자에게 들려주어 어느 프롬프트를 선호하는지 선호도를 조사해야 한다.

인식 문법을 설계할 때 인식 명령어를 기억하는 부담을 줄이기 위해서는 사용성 평가 중에 사용자가 실제로 사용하는 명령어들을 수집해야 한다. 즉, 사용성 평가 도중 사용자가 주어진 작업을 수행하기 위해 발화하는 명령어들

을 수집해야 한다. 이 때 수집된 명령어들을 동일하거나 유사한 명령어들로 분류하고, 각 분류별 빈도수를 측정한다. 또는 미리 사용자가 발화할 것이라고 예상한 명령어 후보들을 선정하고, 사용성 평가 중이나 사용성 평가 종료 후 사용자에게 실제로 발화할 것 같은 명령어 후보가 무엇인지를 질의하여 사용자의 선택을 기록한다. 응답 결과를 분석하여 각 후보들의 빈도수를 측정한다. 마지막으로 빈도수가 높은 명령어들을 음성 인식 명령어로 지정한 후 인식 문법을 설계하면 사용자가 명령어를 기억하는 노력을 감소시키고, 발화하기 자연스러운 명령어 사용을 통해 인식 성공률을 증가시킬 수 있다.

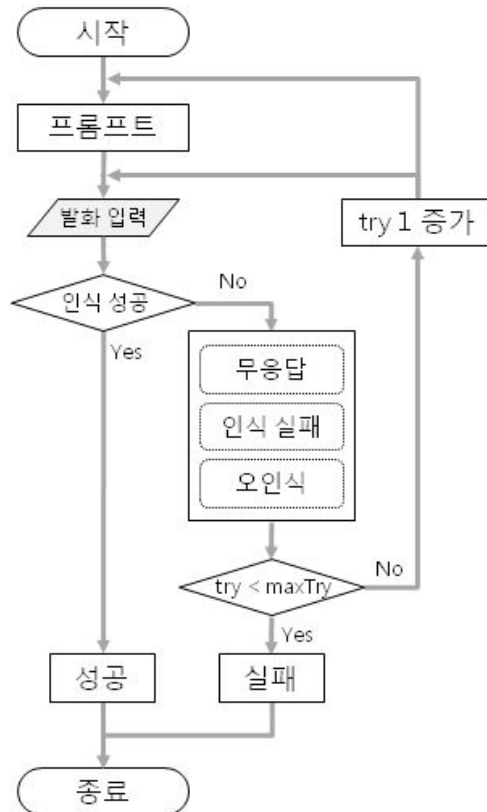
대화 로직을 자연스럽게 설계하기 위해 사용자에게 다양한 대화 흐름을 시뮬레이션함으로써 사용자가 사용을 중도에 포기하지 않고 성공적으로 작업을 완료하는데 적합한 대화 패턴을 수집해야 한다.

본 논문에서는 음성 사용자 인터페이스의 세 가지 구성 요소 중 대화 로직을 제외한 프롬프트와 인식 문법의 사용성 향상을 목표로 하여 설계하기 위한 요소에 초점을 맞춘다.

2) 명령형 음성 인터페이스

명령형 음성 인터페이스란 사용자의 음성 명령을 시스템이 인식하여 인식한 명령어에 상응하는 기능을 수행하도록 하는 인터페이스를 말한다. 사용자의 발화가 입력 모달리티가 되며 시스템의 음성 합성이나 오디오 출력이 출력 모달리티가 된다. 이 때 시스템과 사용자가 대화를 통해 상호작용이 이루어지는데 이를 다이얼로그라고 한다.

명령형 음성 인터페이스의 사용자 중심 설계를 위해서 VoiceXML[8][9][10]을 분석하였다. 분석한 내용을 기초로 하여 명령형 음성 인터페이스의 다이얼로그 모델을 [그림 3-1]과 같이 구성하였다.



[그림 3-1] 다이얼로그 모델

다이얼로그가 시작되면 사용자의 명령을 유도하기 위한 ‘프롬프트’가 출력된다. 이 때 시스템은 주어진 일정 시간 동안 사용자의 발화를 입력받기 위해 대기한다. 이 일정시간은 시스템이 음성 인식을 위해 사용자의 발화를 대기하고 있는 시간으로 다이얼로그 설계 시 설정한다. 시스템에서 인식의 성공과 실패를 판단하는 기준은 인식 결과 점수를 어느 정도까지 성공으로 결정하는가에 따라 달라진다.

사용자의 발화에 따른 인식 성공과 실패에 따른 다이얼로그 흐름은 다음과 같다.

– **인식 성공**

사용자가 주어진 일정 시간 내에 명령어를 정확히 발화하여 시스템이 인식에 성공한 경우 ‘성공’으로 처리되고 다이얼로그는 종료된다.

반면에 인식에 성공하지 않은 경우 ‘무응답’, ‘인식 실패’, ‘오인식’, ‘실패’ 중 한 가지 경우로 처리된다.

– **무응답**

사용자가 주어진 시간 내에 발화하지 못하거나 입력받은 음성 신호의 크기가 작아 시스템이 반응하지 못한다면 ‘무응답’으로 처리된다.

– **인식 실패**

사용자가 명령어를 발화했는데 발음이 부정확하거나 발화한 명령이 시스템이 예상하고 있는 명령어와 일치하는 정도가 낮아 인식 처리를 하지 못한다면 ‘인식 실패’로 처리된다.

– **오인식**

사용자가 발화한 명령이 인식이 되었으나 시스템이 다른 명령어로 인식하면 ‘오인식’으로 처리된다.

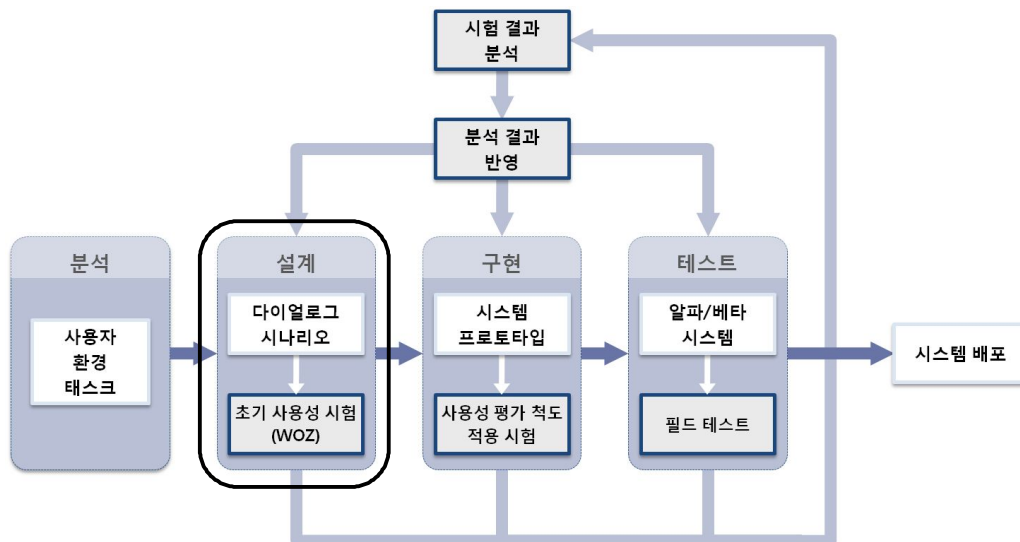
– **실패**

‘무응답’, ‘인식 실패’, ‘오인식’으로 처리된 경우 사용자는 최대 발화 가능 횟수(maxTry)가 넘지 않을 때까지 명령을 재시도할 수 있다. 만약 사용자가 최대 발화 가능 횟수만큼 시도했으나 ‘성공’하지 못한 경우 ‘실패’로 처리되고 다이얼로그는 종료된다.

3) 도구 설계 및 구현 범위

음성 인터페이스를 설계하는 초기 단계에서는 프롬프트, 인식 문법, 대화 로직의 세 가지 구성 요소가 설계된다. 이 때 사용자 중심 설계 방법을 적용하는 경우, 설계 결과물은 사용성 평가를 통해 사용성이 측정되고, 평가 결과를 분석한 내용은 다시 설계에 반영되어 재설계되는 과정을 거친다.

본 논문에서는 사용자 중심 설계 방법을 적용한 음성 인터페이스 설계 과정을 자동화하는 도구를 설계하고 구현하였다. 이 도구는 [그림 3-2]의 ‘설계’ 단계에서 작성된 다이얼로그 시나리오를 바탕으로 ‘초기 사용성 평가’를 하는 과정에서 사용될 것이다.

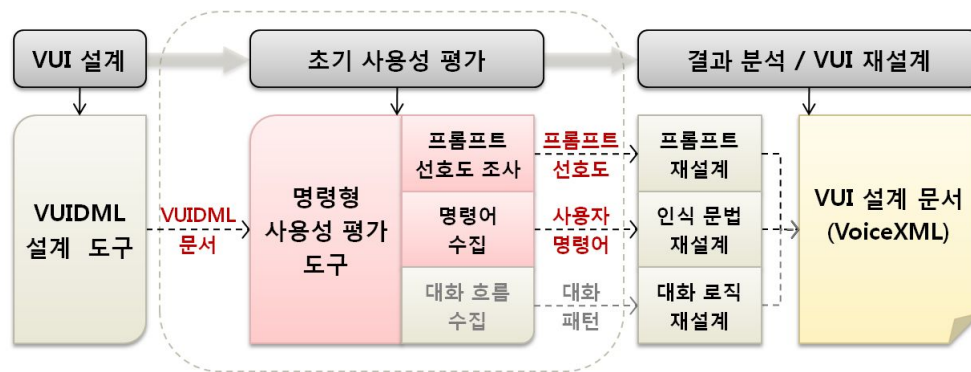


[그림 3-2] 음성 인터페이스의 사용자 중심 설계 과정[7]

초기 사용성 평가는 위자드 오브 오즈 방식으로 진행되어 위자드와 서브젝트가 서로 분리된 공간에서 도구를 통해 의사소통하여 사용성 평가를 실시할 수 있는 구조로 구현된다. 위자드는 다이얼로그 시나리오의 시뮬레이

션을 제어하고 서브젝트의 발화를 관찰하고 기록하는 역할을 맡는다. 서브젝트는 사용성 평가에 참가하는 실제 사용자를 의미하며 위자드가 시뮬레이션하는 다이얼로그에 직접 참여하여 발화하게 된다. 시뮬레이션에 사용되는 다이얼로그 시나리오는 음성 인터페이스 구성 요소들의 사용성을 평가하는 기능을 지원하도록 작성되어야 한다. 또한 음성 사용자 인터페이스의 구성 요소가 설계된 내용을 포함해야 하며 구성 요소가 사용성 평가 대상이 되도록 전용 문서로 작성되어야 한다. 이와 같은 기능을 제공할 수 있는 전용 문서 작성을 위해 이 장의 2절에 Voice User Interface Design Markup Language(VUIDML)을 제안하였다.

음성 사용자 인터페이스를 설계할 때 초기 사용성 평가를 실시하여 사용성을 향상시키는 과정은 [그림 3-3]과 같은 단계로 진행된다.



[그림 3-3] 도구 구현 범위

본 논문의 연구 범위는 다음과 같다. 설계된 음성 사용자 인터페이스를 대상으로 초기 사용성 평가를 실시하기 위해 시뮬레이션용 다이얼로그 시나리오를 작성하기 위한 전용 XML 기반 언어인 VUIDML을 제안하였다. VUIDML로 작성된 문서를 이용하여 초기 사용성 평가 단계에서 다이얼로그를 시뮬레이션한다. 이 때 사용되는 명령형 사용성 평가 도구는 위자드 오브 오즈 방식으로 구성되어 있으며, 시뮬레이션 과정에서 서브젝트를 대

상으로 프롬프트 선호도 조사와 명령어 수집을 실시한다. 도구를 이용하여 선호하는 프롬프트와 사용자 명령어를 초기 사용성 시험 결과로 얻을 수 있다.

VUI 설계 단계에서 사용될 VUIDML 설계 도구, 대화 로직을 설계하기 위한 명령형 사용성 평가 도구의 대화 흐름 시뮬레이션 기능, 초기 사용성 평가 단계의 결과물인 프롬프트, 인식 문법, 대화 로직의 재설계 및 통합을 거쳐 실제 동작이 가능한 VoiceXML과 같은 완성된 VUI 설계 문서를 생성해주는 도구에 대한 연구가 요구된다.

4) 기능 요구 사항

앞에서 정리한 음성 인터페이스의 사용성 향상 및 다이얼로그 모델을 고려하여 설계하고자 하는 도구에서 충족되어야 하는 기능들은 다음과 같다.

- 시나리오 기반 다이얼로그 자동 진행

사용성 평가 시 사용자를 대상으로 시뮬레이션하기 위한 시나리오가 필요하다. 시나리오에는 프롬프트, 인식 문법, 대화 로직의 요소를 사용성 평가할 수 있도록 작성되어야 한다. 또한 프롬프트 출력과 사용자 발화에 따라 다이얼로그가 자동으로 진행되어야 한다.

- 프롬프트 선호도 조사 및 기록

사용자에게 프롬프트 후보 목록들을 들려주고 프롬프트의 선호도를 조사해야 한다. 조사한 결과를 기록하여 사용성 평가 종료 후 수집된 선호도 결과를 분석할 수 있어야 한다.

- 사용자 발화 명령어 수집

인식 문법을 설계하기 위하여 사용자가 실제로 발화하는 명령어들을 수집

해야 한다. 수집을 위해 사용자가 발화한 내용을 오디오 데이터로 캡처하고 파일로 저장해야 한다.

- 다양한 패턴의 다이얼로그 흐름 제공

사용자의 발화에 따라 다양한 패턴으로 음성 인식 결과를 사용자에게 피드백할 수 있어야 한다. 또한 프롬프트 출력 시뮬레이션을 여러 흐름으로 제공하여 사용자가 자연스럽게 느끼는 다이얼로그 흐름을 조사할 수 있어야 한다.

- 위자드 오브 오즈 방식의 음성 인식 시뮬레이션

이 방법을 적용하여 설계 단계의 음성 인식 기반 음성 사용자 인터페이스 시스템의 사용성 평가를 실시할 때, 사용자가 위자드의 존재를 인지하지 못하도록 환경을 구성한다. 실제로 시스템이 완전히 구현된 것처럼 시뮬레이션을 진행하여 사용성 평가 중에 위자드가 사용자의 기대에 응답하고 자연스러운 행위를 유도할 수 있다.

- 사용자 발화 실시간 모니터링

사용자가 발화한 내용을 위자드가 실시간으로 모니터링하여 인식 결과를 선택하여 사용자에게 출력해야 하므로 사용자가 발화한 내용을 실시간으로 모니터링할 수 있어야 한다.

- 로그 기록

위자드와 서브젝트 간에 이루어진 요청 및 응답 결과를 기록할 수 있어야 한다.

- 다이얼로그 시뮬레이션 전용 문서

앞에서 정리된 것처럼 시나리오 기반으로 다이얼로그를 시뮬레이션 하기

위해서는 다이얼로그 시뮬레이션 전용을 위한 문서구조가 필요하다. 이 기능을 지원하기 위한 전용 문서 작성을 위해 VUIDML을 설계하였다.

본 논문에서는 음성 사용자 인터페이스 구성 요소 중 ‘대화 로직’의 설계를 다루고 있지 않으므로, 기능 요구 사항 항목 중 ‘다양한 패턴의 다이얼로그 흐름 제공’은 4 장에서 소개되는 도구의 구현 기능에서 제외되었다.

2. Voice User Interface Design Markup Language

앞 절에서 소개한 명령형 다이얼로그 모델을 기반으로 음성 사용자 인터페이스의 사용성 평가를 위한 Voice User Interface Design Markup Language (VUIDML)를 설계하였다.

1) 기능 및 특징

VUIDML을 이용하여 음성 사용자 인터페이스의 구성 요소 중 프롬프트와 인식 문법을 설계하기 위한 사용성 평가의 시나리오를 작성할 수 있다. VUIDML은 크게 ‘프롬프트 선호도 조사’와 ‘음성 명령어 수집’의 두 가지 기능을 지원한다.

◦ 프롬프트 선호도 조사

사용자에게 다수의 프롬프트를 각각 들려주어 사용자가 선호하는 프롬프트를 직접 선택할 수 있도록 시나리오를 작성할 수 있다. 대화를 시작할 때 뿐만 아니라 음성 인식 시뮬레이션 과정에서 사용자가 접하게 되는 ‘인식 성공’, ‘무응답’, ‘인식 실패’, ‘오인식’, ‘실패’ 상황에서 재생되는 프롬프트들의 선호도 또한 조사하도록 VUIDML 작성이 가능하다.

◦ 음성 명령어 수집

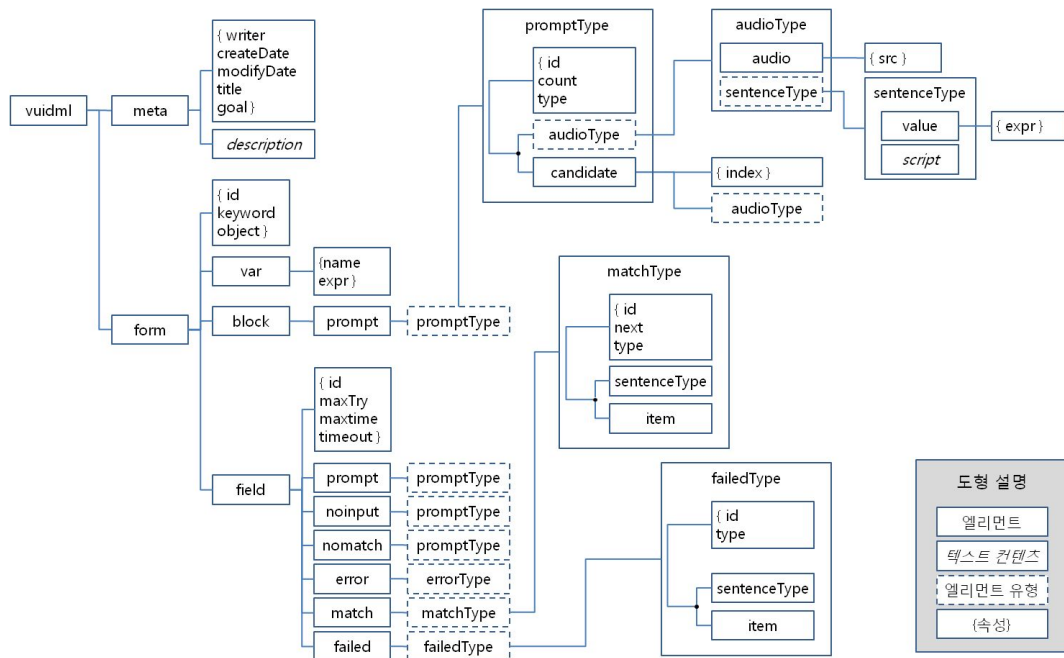
사용자가 기능을 동작시킬 때 자주 발화하는 명령어들을 수집하여 음성 인식 문법을 작성하기 위한 음성 명령어들을 시나리오 시뮬레이션을 통해 자연스럽게 수집할 수 있다.

VoiceXML은 실제 시스템에서 동작하기 위한 다이얼로그 시나리오 작성이 가능한 반면에 VUIDML은 완성된 다이얼로그 시나리오 문서를 설계하

기 위한 사용성 평가 시뮬레이션용 시나리오 작성이 가능하다. 또한 VUIDML 문서는 다이얼로그가 어느 정도 진행되었는지 이력을 파악할 수 있다. 즉, VUIDML 문서는 사용성 평가에 사용되어 원래 설계되어있던 문서보다 사용성이 향상된 VoiceXML 문서를 생성해내기 위해 존재한다.

2) 구성 요소

[그림 3-4]는 VUIDML 을 구성하는 엘리먼트들과 각 엘리먼트들의 속성을 나타낸다.



[그림 3-4] VUIDML 구조

다음은 VUIDML 문서를 구성하는 엘리먼트들과 각 엘리먼트들이 포함할 수 있는 자식 엘리먼트들과 속성들에 대한 설명이다. 자식 엘리먼트 설명 중 [text]항목은 해당 엘리먼트의 텍스트 콘텐츠를 의미한다. 속성에 대한 설명에서 '필수' 이하 항목들은 해당 엘리먼트 작성 시 반드시 값을 지정해

야하는 속성들이고, ‘선택’ 이하 항목들은 작성 시 값을 지정하지 않아도 되는 속성들이다. 예를 들어 <form> 엘리먼트를 설명하는 [표 3-3]에서 설명된 것처럼 ‘id’ 속성은 <form> 엘리먼트를 작성할 때 반드시 값을 지정해 주어야 하고, ‘keyword’, ‘object’ 속성은 VUIDML 설계자가 필요에 따라 값을 지정하거나 지정하지 않아도 된다.

(1) <vuidml> 엘리먼트

VUIDML 문서의 루트 엘리먼트로 문서 내에 한 개만 포함되어야 한다. 이 엘리먼트에 포함시킬 수 있는 자식 엘리먼트와 속성은 [표 3-1]에 나타나있다.

[표 3-1] <vuidml> 엘리먼트

구분	설명
자식 엘리먼트	<ul style="list-style-type: none"> ◦ <meta>: 1 개를 반드시 포함한다. ◦ <form>: 1 개를 반드시 포함한다.
속성	<ul style="list-style-type: none"> • 필수: <ul style="list-style-type: none"> ◦ version: 작성된 문서에 사용된 VUIDML의 버전을 의미한다. 이 속성은 반드시 포함되어야 한다.

[그림 3-5]는 <vuidml> 의 기본 구조를 간략하게 보여준다. <vuidml>은 <meta> 엘리먼트와 <form> 엘리먼트를 한 개씩 포함해야 한다. <meta> 엘리먼트 다음에 <form> 엘리먼트가 오도록 작성한다.

```

<vuidml version="1.0">
  <meta> ... </meta>
  <form> ... </form>
</vuidml>
```

[그림 3-5] <vuidml> 엘리먼트 작성 예

(2) <meta> 엘리먼트

<meta>에는 VUIDML의 작성 정보 및 설계한 시나리오의 제목, 목표, 상세 설명을 포함한다. 이 엘리먼트에 포함시킬 수 있는 자식 엘리먼트와 속성은 [표 3-2]에 나타나있다.

[표 3-2] <meta> 엘리먼트

구분	설명
자식 엘리먼트	<ul style="list-style-type: none"> ◦ [text]: VUIDML 문서에 작성된 다이얼로그 시나리오의 상세 설명을 기술한다. 즉, 설계한 다이얼로그가 시뮬레이션 되었을 때 표현될 내용과 역할 등을 설명한다. ‘title’과 ‘goal’ 속성에 명시한 내용 보다 상세한 설명을 추가한다.
속성	<ul style="list-style-type: none"> • 필수 <ul style="list-style-type: none"> ◦ writer: 문서를 작성한 사람을 의미한다. ◦ createDate: 처음 문서를 작성한 날짜를 의미한다. ‘yyyy-mm-dd’ 와 같은 형식으로 작성한다. (예: 2010년 10월 12일은 ‘2010-10-12’ 로 표기한다.) ◦ modifyDate: 문서를 마지막으로 수정한 날짜를 의미한다. ‘createDate’와 같은 형식으로 작성한다. ◦ title: 시나리오의 제목을 의미한다. ◦ goal: 사용성 평가를 통해 문서의 시나리오에서 달성해야할 목표를 의미한다. 핵심 내용을 간단명료하게 작성한다.

[그림 3-6]은 <meta> 엘리먼트를 XML로 작성한 예를 보여준다. 이 <meta> 엘리먼트로 해당 VUIDML 문서가 ‘홍길동’에 의해 ‘2010-10-11’년 처음 작성되었고 ‘2010-10-15’일에 마지막으로 수정되었다는 것을 알 수 있다.

```

<meta writer="홍길동" createDate="2010-10-11" modifyDate="2010-10-15"
      title="게임 시작" object="게임 시작 명령어 수집">
      게임을 시작하기 위한 명령어를 수집하는 시나리오입니다.
</meta>

```

[그림 3-6] <meta> 엘리먼트

(3) <form> 엘리먼트

<form>은 사용성 평가 대화를 설계한 시나리오의 내용을 포함한다. 하나의 구체적인 목표가 있는 작업의 단위가 된다. 이 엘리먼트에 포함시킬 수 있는 자식 엘리먼트와 속성은 [표 3-3]에 나타나있다.

[표 3-3] <form> 엘리먼트

구분	설명
자식 엘리먼트	<ul style="list-style-type: none"> ◦ <var>: 0 또는 1개 이상 포함한다. ◦ <block>: 0 또는 1개 이상 포함한다. ◦ <field>: 0 또는 1개 이상 포함한다.
속성	<ul style="list-style-type: none"> • 필수 <ul style="list-style-type: none"> ◦ id: <form>의 아이디이다. 문서 내에서 중복되지 않는 값으로 지정한다. • 선택 <ul style="list-style-type: none"> ◦ keyword: <form>의 핵심어를 의미한다. ◦ object: <form>의 구체적인 목적을 의미한다.

[그림 3-7]은 <form> 엘리먼트를 XML로 작성한 예를 보여준다.

```

<form id="form_01" keyword="게임시작명령어" object="명령어수집">
  <var name="sjName" expr="홍길동"/>
  <block id="block_01">...</block>
  <field id="field_01">...</field>
</form>

```

[그림 3-7] <form> 엘리먼트 작성 예

(4) <block> 엘리먼트

<block>은 기능을 설명하는 등의 단순한 출력을 목적으로 사용한다. 또는 프롬프트의 선호도를 조사하기 위한 목적으로 사용한다. 즉, <block> 엘리먼트 내에 다수의 출력 후보 목록을 포함하는 프롬프트를 추가하여 사용자가 선호하는 프롬프트가 무엇인지 조사하도록 설계할 수 있다. 이 엘리먼트에 포함시킬 수 있는 자식 엘리먼트와 속성은 [표 3-4]에 나타나있다.

[표 3-4] <block> 엘리먼트

구분	설명
자식 엘리먼트	<ul style="list-style-type: none"> ◦ <prompt>: 1 개 이상 포함한다. <block> 내의 <prompt> 엘리먼트가 다수인 경우 작성된 순서대로 통합되어 출력된다.
속성	<ul style="list-style-type: none"> • 필수 <ul style="list-style-type: none"> ◦ id: <block>의 아이디이다. 문서 내에서 중복되지 않는 값으로 지정한다.

[그림 3-8]은 <block> 엘리먼트를 XML로 작성한 예를 보여준다. <prompt> 엘리먼트를 이용하여 사용자에게 시스템의 기능을 설명하고 있다.

```
<block id="block_01">
  <prompt id="ppt_01">
    지금 준비되어있는 게임은 낱말 맞추기입니다.
  </prompt>
</form>
```

[그림 3-8] <block> 엘리먼트 작성 예

(5) <field> 엘리먼트

<field>는 사용자의 음성 명령을 받아들여야 하는 경우 사용한다. <prompt>, <noinput>, <nomatch>, <error>, <match>, <failed>를 포함한다. 이 엘리먼트에 포함시킬 수 있는 자식 엘리먼트와 속성은 [표 3-5]에 나타나있다.

[표 3-5] <field> 엘리먼트

구분	설명
자식 엘리먼트	<ul style="list-style-type: none"> ◦ <prompt>: 1 개 이상 반드시 포함한다. ◦ <noinput>: 1 개 이상 반드시 포함한다. ◦ <nomatch>: 1 개 이상 반드시 포함한다. ◦ <error>: 1 개 이상 반드시 포함한다. ◦ <match>: 1 개를 반드시 포함한다. ◦ <failed>: 1 개를 반드시 포함한다.
속성	<ul style="list-style-type: none"> • 필수 <ul style="list-style-type: none"> ◦ id: <field>의 아이디이다. 문서 내에서 중복되지 않는 값으로 지정한다. • 선택 <ul style="list-style-type: none"> ◦ maxTry: 사용자가 음성 명령을 시도할 수 있는 최대 횟수이다. 이 횟수 내에 인식에 성공해야 <match>가 출력된다. 기본 값은 '3'으로 지정하지 않으면 3으로 지정된다. ◦ maxTime: 시스템이 음성 인식을 대기하고 있는 시간이다. 단위는 밀리세컨드(1/1000초)로 지정한다. ◦ recPath: 사용자가 발화했을 때 발화한 오디오 데이터가 파일로 저장되는 경로이다. 파일명은 prompt 아이디와 try 수를 조합하여 자동으로 지정한다.(유형제시) 절대 경로 또는 상대 경로 형태로 입력한다.

[그림 3-9]는 <field> 엘리먼트를 XML로 작성한 예를 보여준다. <prompt> 엘리먼트를 이용하여 사용자에게 시스템의 기능을 설명하고 있다. maxTry 값을 지정하지 않았으므로 기본 값인 '3'이 적용된다. 인식 대기시간은 최대 3초가 되고 사용자가 발화한 내용은 동작되는 시스템을 기준으로 './'에 저장된다.

```

<field id="field_01" maxTime="3000" recPath="./">
  <prompt id="ppt_02">
    제시된 게임 중 하나를 골라 말씀해주세요.
  </prompt>
  ...
  <failed id="failed_01">처음 메뉴로 돌아갑니다.</failed>
</form>

```

[그림 3-9] <block> 엘리먼트 작성 예

(6) <prompt> 엘리먼트

<prompt>는 음성 합성으로 출력할 문장이나 녹음된 오디오 파일을 포함한다. 한 문장만을 포함하는 'simple' 유형과 다수의 후보 문장들을 포함하는 'alternative' 유형이 있다. 'simple' 유형인 경우에는 표 6의 [text], <audio>, <value> 엘리먼트들이 복합적으로 한 번의 출력을 구성할 수 있다. 예를 들어 효과음과 임의의 값, 합성할 문장을 하나의 프롬프트로 출력하기 위해 <audio>, <value>, [text]를 이용하여 <prompt>를 구성할 수 있다. 'alternative' 유형의 경우에는 <candidate> 엘리먼트를 이용하여 2개 이상의 후보 프롬프트를 추가해야한다. 이 엘리먼트에 포함시킬 수 있는 자식 엘리먼트와 속성은 표 6에 나타나있다.

[표 3-6] <prompt> 엘리먼트

구분	설명
자식 엘리먼트	<ul style="list-style-type: none"> ◦ [text]: 출력할 문장을 입력한다. 'simple'인 경우에만 가능하다. ◦ <audio>: 0 또는 1개 이상 포함한다. 'simple'인 경우에만 가능하다. ◦ <value>: 0 또는 1개 이상 포함한다. 'simple'인 경우에만 가능하다. ◦ <candidate>: 'simple'인 경우에는 0 개, 'alternative'인 경우에는 2개 이상 포함한다.
속성	<ul style="list-style-type: none"> • 필수 <ul style="list-style-type: none"> ◦ id: <prompt>의 아이디이다. 문서 내에서 중복되지 않는 값으로 지정한다. • 선택 <ul style="list-style-type: none"> ◦ count: 출력할 순서를 의미한다. ◦ type: 프롬프트 구성 유형을 의미한다. 'simple'과 'alternative'를 값으로 지정할 수 있으며 기본 값은 'simple'이다. 'type' 속성을 별도로 지정하지 않으면 'simple'로 설정된다.

[그림 3-10]은 <prompt> 엘리먼트를 XML로 작성한 예를 보여준다.

```

<prompt type="alternative">
  <candidate index="1">
    시작할 게임의 이름을 말씀해 주세요. </candidate>
  <candidate index="2"> 게임을 시작해주세요. </candidate>
  <candidate index="3"> 도움말이 시작됩니다. </candidate>
</prompt>

```

[그림 3-10] <prompt> 엘리먼트 작성 예

(7) <noinput>, <nomatch>, <error> 엘리먼트

<noinput>, <nomatch>, <error> 엘리먼트들은 사용자가 발화했을 때 실패한 경우 사용자에게 프롬프트하기 위한 용도로 사용된다. 무응답을 의미하는 <noinput>, 인식 실패인 경우 <nomatch>, 오인식인 경우 <error>를 출력한다. 시뮬레이션시 카운트되고 있는 시도 횟수가 <field> 엘리먼트에 지정한 'maxTry' 값을 초과하면 <failed> 엘리먼트의 내용이 출력된다. 이 엘리먼트들은 포함 가능한 엘리먼트들과 속성들이 <prompt>의 것과 동일하다([표 3-6] 참고).

(8) <match> 엘리먼트

<field>에서 사용자의 발화가 인식된 경우 출력된다. <match> 에 도달했다는 것은 사용자가 <field>의 목적을 달성했다는 것을 의미한다. 이 엘리먼트에 포함시킬 수 있는 자식 엘리먼트와 속성은 [표 3-7]에 나타나있다.

[표 3-7] <match> 엘리먼트

구분	설명
자식 엘리먼트	<ul style="list-style-type: none">◦ [text]: 출력할 문장을 입력한다. 'simple'인 경우에만 가능하다.◦ <audio>: 0 또는 1개 이상 포함한다. 'simple'인 경우에만 가능하다.◦ <value>: 0 또는 1개 이상 포함한다. 'simple'인 경우에만 가능하다.◦ <candidate>: 'simple'인 경우에는 0 개, 'alternative'인 경우에는 2개 이상 포함한다.
속성	<ul style="list-style-type: none">• 필수<ul style="list-style-type: none">◦ id: <match>의 아이디이다. 문서 내에서 중복되지 않는 값으로 지정한다.• 선택<ul style="list-style-type: none">◦ type: 프롬프트 구성 유형을 의미한다. 'simple'과 'alternative'를 값으로 지정할 수 있으며 기본 값은 'simple'이다. 'type' 속성을 별도로 지정하지 않으면 'simple'로 설정된다.

[그림 3-11]은 <match> 엘리먼트를 XML로 작성한 예를 보여준다.

```
<match type="alternative">
  <candidate index="1">
    '날말 맞추기가'가 선택되었습니다. '날말 맞추기'를 실행합니다.
    잠시만 기다려주세요.
  </candidate>
  <candidate index="2">
    '날말 맞추기'를 실행합니다. 잠시만 기다려주세요.
  </candidate>
</match>
```

[그림 3-11] <match> 엘리먼트 작성 예

(9) <failed> 엘리먼트

<field>에서 사용자가 최대 인식 가능 회수를 초과해 음성 명령을 시도한 경우 출력된다. <failed>에 도달했다는 것은 <match>와 반대로 <field>의 목적을 달성하는데 실패했다는 것을 의미한다. 이 엘리먼트에 포함시킬 수 있는 자식 엘리먼트와 속성은 [표3-8]에 나타나있다.

[표 3-8] <failed> 엘리먼트

구분	설명
자식 엘리먼트	<ul style="list-style-type: none"> ◦ [text]: 출력할 문장을 입력한다. 'simple'인 경우에만 가능하다. ◦ <audio>: 0 또는 1개 이상 포함한다. 'simple'인 경우에만 가능하다. ◦ <value>: 0 또는 1개 이상 포함한다. 'simple'인 경우에만 가능하다. ◦ <candidate>: 'simple'인 경우에는 0 개, 'alternative'인 경우에는 2개 이상 포함한다.
속성	<ul style="list-style-type: none"> • 필수 <ul style="list-style-type: none"> ◦ id: <failed>의 아이디이다. 문서 내에서 중복되지 않는 값으로 지정한다. • 선택 <ul style="list-style-type: none"> ◦ type: 프롬프트 구성 유형을 의미한다. 'simple'과 'alternative'를 값으로 지정할 수 있으며 기본 값은 'simple'이다. 'type' 속성을 별도로 지정하지 않으면 'simple'로 설정된다.

[그림 3-12]는 <failed> 엘리먼트를 XML로 작성한 예를 보여준다. 사용자가 유효한 음성 명령 시도 횟수를 초과하였으므로 시스템의 화면이 처음으로 돌아간다는 안내를 사용자에게 프롬프트 한다.

```
<failed>
```

사용 방법을 숙지한 후에 다시 시도해주세요. 처음 화면으로 이동합니다.

```
</failed>
```

[그림 3-12] <failed> 엘리먼트 작성 예

(10) <candidate> 엘리먼트

<prompt>, <noinput>, <nomatch>, <error>, <match>, <failed> 엘리먼트에서 유형이 'alternative'인 경우 출력할 후보 프롬프트들은 <candidate>로 추가한다. 이 엘리먼트에 포함시킬 수 있는 자식 엘리먼트와 속성은 [표 3-9]에 나타나있다.

[표 3-9] <candidate> 엘리먼트

구분	설명
자식 엘리먼트	<ul style="list-style-type: none">◦ [text]: 출력할 문장을 입력한다. 'simple'인 경우에만 가능하다.◦ <audio>: 0 또는 1개 이상 포함한다. 'simple'인 경우에만 가능하다.◦ <value>: 0 또는 1개 이상 포함한다. 'simple'인 경우에만 가능하다.
속성	<ul style="list-style-type: none">• 필수◦ index: 후보 프롬프트들의 순서를 의미한다.

(11) <audio> 엘리먼트

음성 합성 대신 사용할 오디오 파일은 <audio>를 사용하여 추가한다. 합성할 문장과 <value>와 조합하여 프롬프트를 구성하거나 단독으로 쓰일 수 있다. 이 엘리먼트에 포함시킬 수 있는 자식 엘리먼트와 속성은 [표 3-10]에 나타나있다.

[표 3-10] <audio> 엘리먼트

구분	설명
자식 엘리먼트	자식노드를 포함하지 않는다.
속성	<ul style="list-style-type: none">필수<ul style="list-style-type: none">src: 재생할 오디오 파일의 경로를 지정한다. 절대 경로 또는 상대경로로 확장자까지 지정한다.

[그림 3-13]은 <audio> 엘리먼트를 XML로 작성한 예를 보여준다. 두 개의 알림음에 대하여 사용자 선호도와 적합성을 알아보기 위해 ‘alternative’ 유형인 <prompt> 엘리먼트 내에 <candidate> 엘리먼트를 이용하여 두 개의 비언어 오디오가 삽입되도록 설계되어있다. 첫 번째 <audio> 엘리먼트를 선택하면 ‘./start-01.wav’파일이 재생되고, 두 번째 <audio> 엘리먼트를 선택하면 ‘./start-02.wav’파일이 재생됨을 의미한다.

```
<block id="block_02">
  <prompt id="ppt_03" type="alternative">
    <candidate index="1">
      <audio src="./start-01.wav"/>
    </candidate>
    <candidate index="2">
      <audio src="./start-02.wav"/>
    </candidate>
  </prompt>
</block>
```

[그림 3-13] <failed> 엘리먼트 작성 예

(12) <var>, <value> 엘리먼트

<var> 는 일종의 변수를 지정하는 목적으로 사용되고 <value>는 지정된 변수를 사용하는 목적으로 사용되는 엘리먼트이다. VUIDML 문서 내에서 어떤 프롬프트의 문장이나 단어가 시뮬레이션 중에 동적으로 변경해 적용시키고자 할 때 사용한다. 또는 같은 용어나 단어가 문서의 여러 부분에서 사용될 때 값의 변경을 쉽게 하고자 하는 용도로도 사용할 수 있다. 이 엘리먼트에 포함시킬 수 있는 자식 엘리먼트와 속성은 [표 3-11]에 나타나있다.

[표 3-11] <var> 엘리먼트

구분	설명
자식 엘리먼트	자식노드를 포함하지 않는다.
속성	<ul style="list-style-type: none"> • 필수 <ul style="list-style-type: none"> ◦ name: 고유한 값을 입력한다. <value> 엘리먼트의 ‘expr’ 속성에 참조된다. ◦ expr: 기본 설정 값을 입력한다.

<value>의 ‘expr’ 속성에 지정된 값은 <var> 엘리먼트의 ‘name’ 값을 참조하는 참조된 값이다.

[표 3-12] <value> 엘리먼트

구분	설명
자식 엘리먼트	자식노드를 포함하지 않는다.
속성	<ul style="list-style-type: none"> • 필수 <ul style="list-style-type: none"> ◦ expr: var 엘리먼트의 ‘name’ 속성 값을 참조한다.

[그림 3-14]는 <var>와 <value> 엘리먼트가 사용된 예이다.

```
<form>
  <var name="sbjName" expr="홍길동"/>
  <block>
    <prompt id="ppt_04">
      <candidate index="1">
        <value expr="sbjName">님 안녕하세요.</candidate>
      <candidate index="2">
        <value expr="sbjName">씨 반가워요.</candidate>
      </prompt>
    </block>
  </form>
```

[그림 3-14] <var> 와 <value> 엘리먼트 사용 예

3. VUIDML 문서 활용 예

VUIDML 문법으로 사용성 평가용 다이얼로그 시나리오 문서를 작성할 수 있다. [그림 3-15]의 문서는 <block> 엘리먼트를 이용하여 프롬프트 선택도를 조사하는 목적으로 설계되었다.

```
<vuidml version="1.0">
  <meta writer="홍길동" createDate="2010-10-11" modifyDate="2010-10-15"
        title="게임 시작 명령어 수집" goal="게임 시작">
    게임을 시작하기 위한 명령어를 수집한다.
  </meta>
  <form id="form1" keyword="게임 시작" object="게임 시작 프롬프트 선택도">
    <var name="game" expr="날말 맞추기"/>
    <block id="bk_0">
      <prompt id="ppt_01" type="alternative">
        <candidate index="1">
          <value expr="game"/> 시작 </candidate>
        <candidate index="2">
          <value expr="game"/>을 시작하겠습니다. </candidate>
        <candidate index="3">
          <value expr="game"/>를 선택하셨습니다.
          <value expr="game"/>를 시작하겠습니다. </candidate>
        </prompt>
      </block>
    </form>
  </vuidml>
```

[그림 3-15] VUIDML 문서 작성 예1

문서가 해석될 때 <var> 엘리먼트의 expr 속성 값이 <prompt> 하위의 <value> 엘리먼트와 변경된다. 즉, <prompt> 의 각 <candidate> 엘리먼트들은 ‘날말 맞추기 시작’, ‘날말 맞추기를 시작하겠습니다.’, ‘날말 맞추기를 선택하셨습니다. 날말 맞추기를 시작하겠습니다.’의 텍스트 문장으로 변경된다. 변경된 문장을 사용자에게 각각 한 개씩 들려준 후 어떤 프롬프트가 듣는데 자연스러운지를 선호도를 조사할 수 있다.

[그림 3-16]은 VUIDML의 문법을 사용하여 명령어 수집을 위한 음성 인터페이스 사용성 평가용 시나리오를 작성한 것이다. 첫 번째 <prompt>는 사용자의 첫 번째 시도 시 출력되며 세 개의 후보 프롬프트를 포함한다. 두 번째 <prompt>는 한 개의 프롬프트를 포함하고, ‘count’ 속성이 2이므로 두 번째 시도부터 최종 시도까지 ‘ddl.wav’파일이 재생된다. 사용자가 프롬프트를 듣고 발화를 하면 음성 인식 피드백을 사용자에게 제공하기 위해, 발화 상태에 따라 <noinput>, <nomatch>, <error>, <match>, <failld> 를 선택하여 프롬프트하게 된다. <error> 의 경우 두 개의 후보 프롬프트 문장을 포함하는데, 다양한 다이얼로그 흐름을 위해 1번 <candidate> 엘리먼트와 2번 <candidate> 엘리먼트의 문장이 경우에 따라 다르게 선택되어 출력된다.

이처럼 목적에 맞게 VUIDML을 작성하고, 이 문서에 맞추어 음성 인터페이스가 다양한 대화 흐름으로 시뮬레이션 되도록 시나리오를 설계할 수 있다.

```

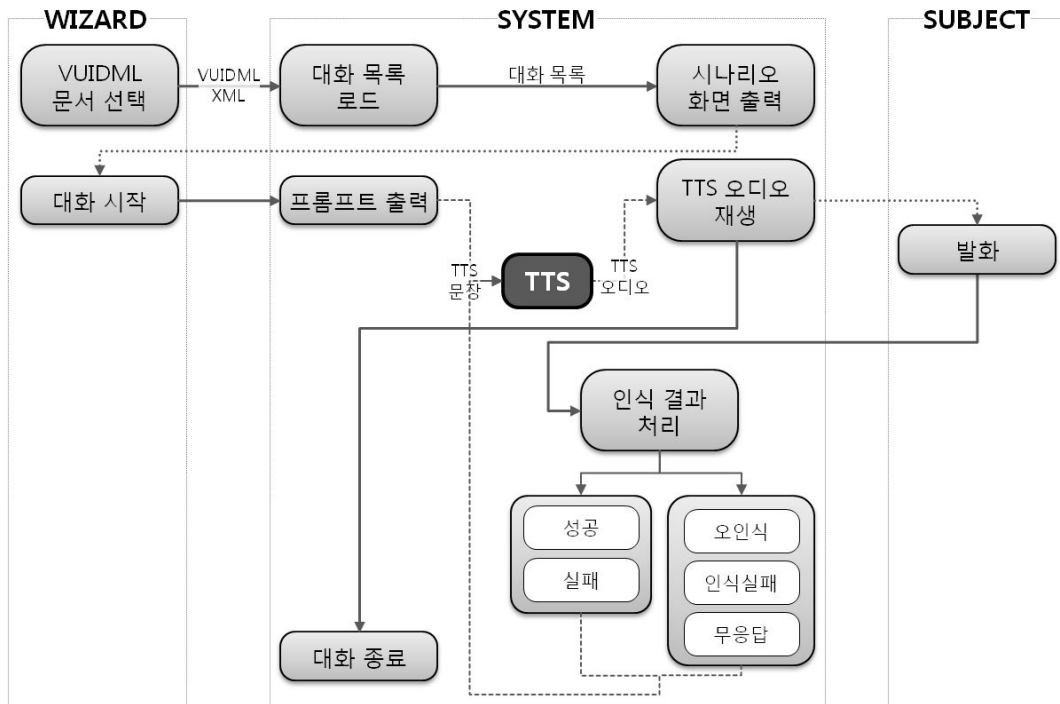
<vuidml version="1.0">
<meta writer="홍길동" createDate="2010-10-11" modifyDate="2010-10-15"
      title="게임 시작 명령어 수집" goal="게임 시작">
  게임을 시작하기 위한 명령어를 수집한다.
</meta>
<form id="form2" keyword="게임 시작" object="시작 명령어 수집">
  <field id="fd_0" maxTry="3" inputType="recording" maxTime="5000">
    <prompt type="alternative">
      <candidate index="1">
        시작할 게임의 이름을 말씀해주세요. </candidate>
      <candidate index="2"> 시작할 게임을 말씀해주세요. </candidate>
      <candidate index="3"> 게임을 시작해주세요. </candidate>
    </prompt>
    <prompt count="2" type="simple">
      <audio src="noinput.wav"/> </prompt>
    <noinput type="simple"> 시간이 초과되었습니다. </noinput>
    <noinput type="simple" count="3">
      유효한 입력 신호가 없습니다. 기기 상태를 점검해주세요. </noinput>
    <nomatch> 명령어를 다시 한 번 말씀해주세요. </nomatch>
    <error type="alternative">
      <candidate index="1"> 처음 화면으로 이동합니다. </candidate>
      <candidate index="2"> 도움말이 시작됩니다. </candidate>
    <match> 성함 확인이 완료되었습니다. </match>
    <failed> 사용법을 숙지하신 후 다시 시작해주세요. 대화를 종료합니다.
    </failed>
  </field>
</form>
</vuidml>

```

[그림 3-16] VUIDML 문서 작성 예2

4. VUIDML을 이용한 사용성 평가 과정

VUIDML을 이용하여 작성한 다이얼로그 시나리오의 흐름으로 시뮬레이션 된다.



[그림 3-17] VUIDML 시뮬레이션 과정

- VUIDML 문서 선택

위자드(WIZARD)가 시뮬레이션하고자 하는 VUIDML문서를 선택한다.

- 대화 목록 로드

선택한 VUIDML 문서 내에 속해있는 대화 목록(프롬프트)들이 로드된다. 이 때 대화 목록은 <form> 엘리먼트 내의 <block> 또는 <field> 엘

리먼트 가 포함되어있는 순서대로 구성되어 출력된다.

- 대화 시작, 프롬프트 출력

위자드가 대화를 시작하면 위자드 시스템에서는 초기 프롬프트의 재생을 위한 작업이 이루어진다. 위자드 시스템은 프롬프트에 포함된 텍스트 콘텐츠를 TTS로 전송한다. TTS는 전송 받은 텍스트를 오디오 데이터로 변환해 다시 위자드 시스템으로 전송해 준다.

- TTS 오디오 재생

이 변환된 TTS 오디오가 위자드에서 서브젝트 시스템으로 전송되면 서브젝트 시스템은 서브젝트가 들을 수 있도록 프롬프트를 재생해 준다. 여기서 서브젝트는 시뮬레이션에 참가하고 있는 사용자를 의미한다.

- 발화

서브젝트는 프롬프트를 듣고 내용에 따라 발화한다.

- 인식 결과 처리

서브젝트를 관찰하고 있던 위자드는 발화 내용에 따라 인식 결과를 처리한다. 앞 절에서 설명한 다이얼로그 모델의 흐름에 따라 위자드는 ‘오인식’, ‘인식실패’, ‘무응답’, ‘성공’, ‘실패’를 결정하고 서브젝트에 인식 결과를 프롬프트한다. 이 때 선택된 인식 결과에 따라 프롬프트는 다르게 출력된다.

- 대화 종료

설계된 내용에 따라 ‘성공’하거나 ‘실패’에 이르게 되면 대화는 종료된다.

IV. 명령형 음성 사용자 인터페이스의 사용성 평가를 위한 도구 구현

1. 구현 환경

명령형 음성 인터페이스의 설계를 위한 사용성 평가 도구를 구현한 환경은 [표 4-1]과 같다.

[표 4-1] 구현 환경

구분		내용
하드웨어	CPU	Intel Core2 Duo T7800 2.6GHz
	메모리	4GB
	HDD 용량	200GB
	사운드 카드	온 보드 Realtek High Definition Audio드라이버
	네트워크 환경	유선 LAN 100MB
소프트웨어	운영체제	윈도우 7 엔터프라이즈 32bit
	프로그래밍 언어	Java SDK 1.6.0_22
	프로그래밍 도구	Netbeans 6.9.1
	XML 파서	Java DOM
	음성 합성기	Voiceware TTS server

2. 사용 환경

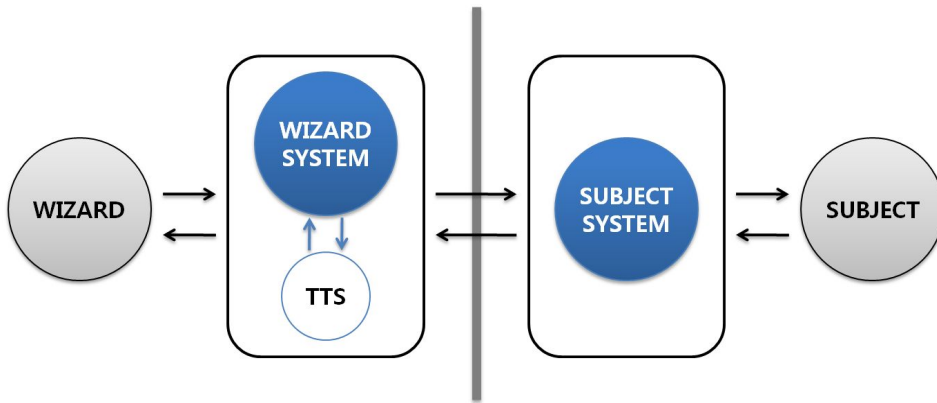
명령형 음성 인터페이스의 설계를 위한 사용성 평가 도구를 사용하기 위해서는 위자드 오브 오즈 방식의 사용성 평가 진행을 위한 분리된 공간과 위자드 측에서 사용할 PC 1대와 서브젝트 측에서 사용할 PC 1대가 각각 필요하다. 각 PC의 권장 사양은 [표 4-2]와 같다.

[표 4-2] 사용 권장 사양

구분		권장 사양
하드웨어	CPU	Intel Pentium 4
	메모리	1GB이상
	HDD 용량	10GB이상
	사운드 카드	재생 및 녹음 지원
	네트워크 환경	인터넷 접속 필요
소프트웨어	운영체제	윈도우 XP 32bit 이상
	JRE	JRE 6 이상 설치
	음성 합성기	Voiceware TTS server

3. 시스템 구성

시스템은 [그림 4-1]에 보이는 것처럼 크게 위자드가 사용하게 될 위자드 시스템과 서브젝트가 사용하게 될 서브젝트 시스템으로 구성된다. 이 두 개의 시스템은 위자드 오브 오즈의 방식에 따라 서로 분리된 환경에서 운영된다.



[그림 4-1] 도구 개념도

위자드(WIZARD)는 VUIDML 문서를 이용하여 다이얼로그를 시뮬레이션하고 서브젝트를 관찰하는 역할을 한다. 시뮬레이션을 위해서 VUIDML의 프롬프트 텍스트를 오디오로 변환해줄 TTS(Text to Speech) 모듈이 동작되어야 한다.

서브젝트(SUBJECT)는 위자드가 시뮬레이션하는 시나리오에 따라 발화하여 명령형 대화를 진행한다.

위자드 시스템(WIZARD SYSTEM)은 위자드에 의해 프롬프트를 전송하고 서브젝트의 발화를 모니터하고 데이터를 저장한다.

TTS는 위자드 시스템으로부터 전송받은 텍스트를 오디오로 변환하여 다시 위자드 시스템으로 전송한다.

- XML 관리자

위자드가 VUIDML 문서를 선택하면 XML 관리자는 문서를 로드하고 XML 파싱 과정을 수행한다. XML 파싱을 통해 다이얼로그 객체로 엘리먼트들의 정보를 저장한다.

- 다이얼로그 객체

다이얼로그 객체는 프롬프트 출력을 위한 프롬프트 객체를 포함한다. 다이얼로그에 설정해야 할 변수 값이 있다면 위자드로부터 변수 값을 입력 받아야 한다. 위자드에 의해 다이얼로그 시뮬레이션을 위한 설정 값을 입력받고 프롬프트를 출력할 루트를 검색하여 설정한다.

- 프롬프트 객체

프롬프트 객체는 위자드가 프롬프트 전송을 요청했을 때 TTS 관리자에게 자신이 저장하고 있는 TTS 텍스트를 전달하여 오디오 파일 변환을 요청한다. 오디오 파일 변환이 완료되면 전송 관리자에게 오디오 데이터 전송을 요청한다.

- TTS 관리자

프롬프트 객체의 요청에 의해 TTS 서버로 TTS 변환을 요청한다. 변환된 오디오 데이터가 전송되면 전송된 데이터를 파일로 저장하고 경로를 프롬프트 객체에 반환한다.

- 전송 관리자

프롬프트 객체의 요청에 의해 오디오 데이터를 전송한다. 전송할 서브젝트 시스템의 상태를 점검하고 전송을 시작한다.

- 캡처 관리자

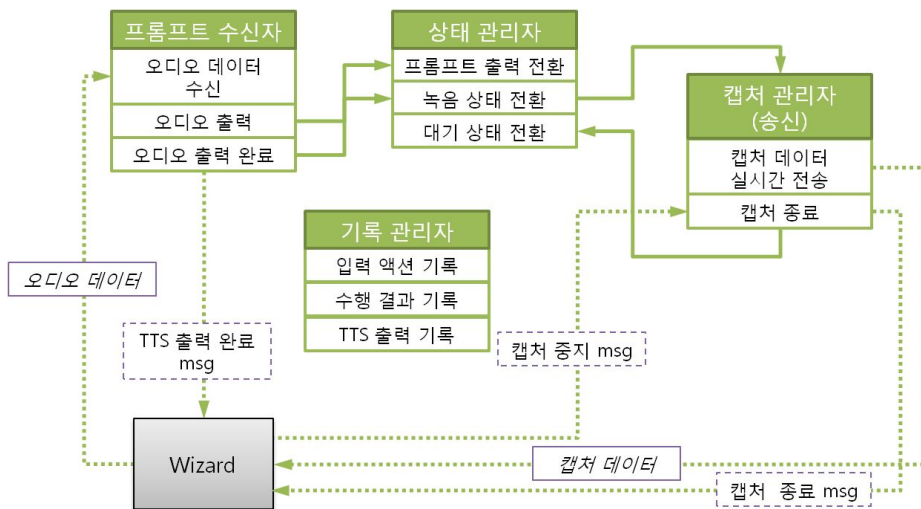
서브젝트 시스템으로부터 TTS 전송이 완료되었다는 메시지를 수신하

면 캡처 관리자는 서브젝트 시스템에서 전송하는 발화 오디오 데이터 수신을 시작한다. 동시에 캡처 관리자는 수신된 데이터를 실시간으로 위자드가 들을 수 있도록 재생한다. 위자드의 캡처 종료 요청에 의하여 수신을 종료하고 캡처 중지 메시지를 서브젝트 시스템으로 전송한다. 서브젝트 시스템으로부터 캡처 종료 메시지를 수신하면 수신된 오디오 데이터를 파일로 저장한다.

◦ 기록 관리자

위자드를 통해 입력된 액션을 기록하고 다이얼로그 시뮬레이션 결과를 기록한다.

[그림 4-3]은 서브젝트가 사용하게 될 서브젝트 시스템의 구조를 나타낸다. 사용성 평가 중에 서브젝트는 실제로 서브젝트 시스템을 제어하지는 않는다. 이 시스템은 서브젝트에게 프롬프트를 출력하고 서브젝트의 발화를 캡처하여 위자드 시스템으로 전송하는 기능을 한다.



[그림 4-3] 서브젝트 시스템

- 프롬프트 수신자

위자드 시스템으로부터 오디오 데이터 전송이 시작되면 프롬프트 수신자가 이를 수신하여 오디오로 출력한다. 오디오 출력이 완료되면 위자드 시스템에게 TTS 출력 완료 메시지를 전송한다.

- 기록 관리자

TTS 출력 정보와 동작된 입력 액션 및 수행 결과를 기록한다.

- 캡처 관리자

시스템이 녹음 상태로 전환되면 서브젝트의 발화 내용 캡처를 시작하고 이 때 캡처된 오디오 데이터를 실시간으로 위자드 시스템으로 전송한다.

위자드 시스템으로부터 캡처 중지 메시지가 수신되면 캡처 관리자는 캡처를 종료하고 위자드 시스템으로 캡처 종료 메시지를 보낸다.

- 상태 관리자

시스템 동작에 따라 ‘프롬프트 출력’, ‘녹음 중’, ‘대기 상태’로 시스템의 상태를 전환한다.

4. VUIDML 파싱 절차

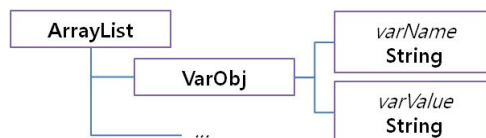
위자드가 VUIDML 형식으로 작성된 XML 문서를 열어 선택된 문서가 로드되는 경우, 또는 <var> 엘리먼트의 속성 값을 변경하기 위한 명령을 내린 경우에 VUIDML을 파싱하기 위한 모듈이 동작된다. 동작되는 순서는 다음과 같다.

1) Document 인스턴스 생성 및 저장

로드된 문서는 DOM 트리 구조로 메모리에 저장된다. DOM 구조의 정보를 파싱하여 Document 인스턴스 객체로 저장된다.

2) <var> 엘리먼트 추출

Document 인스턴스 객체에서 <var> 엘리먼트를 추출하여 [그림 4-4]와 같이 VarObj 객체로 저장된다. VarObj의 문자열 변수인 'varName'에 'name' 속성 값이 지정되고, 다른 문자열 변수인 'varValue'에 'expr' 속성 값이 지정된다. VarObj는 <var> 엘리먼트 수만큼 생성되어 ArrayList에 저장된다.



[그림 4-4] VarObj 오브젝트

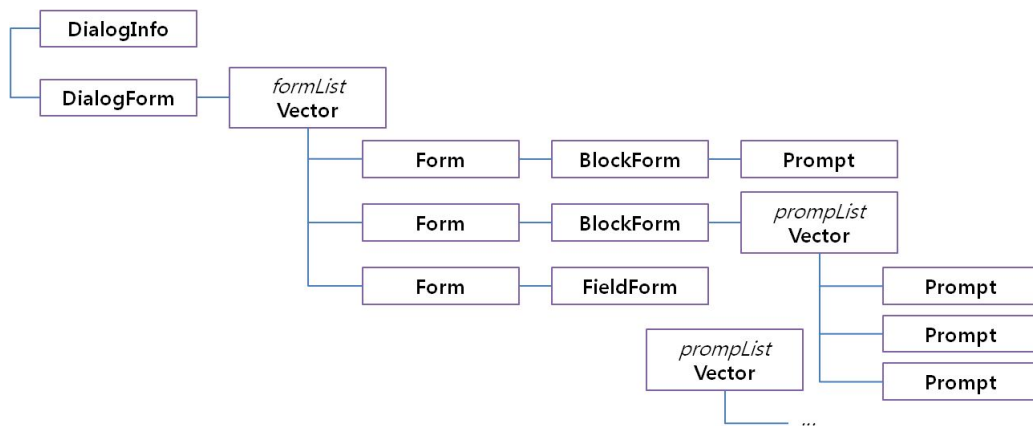
3) <var> 엘리먼트 값 위자드 지정

VarObj를 저장한 ArrayList를 매개변수로 이용하여 위자드에게 <var> 엘리먼트의 값을 변경할 수 있는 대화상자를 띄운다. 이 때 위자드가 값을

수정하면 해당 항목이 참조하는 VarObj 객체의 멤버 변수인 'varValue'의 값이 변경된다.

4) 다이얼로그 컨텍스트 객체 생성

VUIDML 문서의 엘리먼트 정보는 다이얼로그 컨텍스트 객체로 변환되어 저장된다. 저장된 구조는 [그림 4-5]와 같다.



[그림 4-5] 다이얼로그 컨텍스트 객체 구조

- DialogInfo 객체

Document 인스턴스 객체에서 검색된 <meta> 엘리먼트의 정보는 DialogInfo 객체로 저장된다.

- DialogForm 객체

Document 인스턴스 객체에서 검색된 <form> 엘리먼트의 정보는 DialogForm 객체로 저장된다.

- BlockForm 객체

DialogForm 생성시 <form> 엘리먼트 하위 요소인 <block> 엘리먼트의 정보는 BlockForm 객체로 저장된다. 생성된 객체는 DialogForm의

‘formList’ Vector 에 삽입된다.

- FieldForm 객체

DialogForm 생성시 <form> 엘리먼트 하위 요소인 <Field> 엘리먼트의 정보는 FieldForm 객체로 저장된다. 성된 객체는 DialogForm의 ‘formList’ Vector 에 삽입된다.

- Prompt 객체

<block> 엘리먼트 하위의 <prompt> 엘리먼트 정보는 Prompt 객체로 저장되어 해당 BlockForm 객체의 ‘promptList’ Vector에 추가되고, <field> 엘리먼트 하위의 <prompt>, <noinput>, <nomatch>, <error>, <failed>, <match> 엘리먼트들도 <block> 엘리먼트 하위의 <prompt> 엘리먼트와 마찬가지로 해당 FieldForm 객체의 ‘promptList’ Vector 에 Prompt 객체로 변환되어 삽입된다.

5) Prompt 객체의 TTS 텍스트 저장

<prompt> 엘리먼트가 Prompt 객체로 변환될 때 <prompt> 엘리먼트 하위의 <value>, <audio> 엘리먼트와 텍스트 요소들은 TTS 변환을 위한 텍스트로 재구성되어야 한다.

- <value> 엘리먼트

<prompt> 하위 요소 탐색 중 <value> 엘리먼트를 만나면 ‘expr’ 속성 값을 추출하고, VarObj 가 저장된 ArrayList에서 추출한 ‘expr’ 값과 일치하는 ‘varName’ 멤버 변수를 갖는 VarObj를 찾아 ‘varValue’ 멤버 변수 값을 가져온다. 가져온 값을 해당 Prompt 객체의 TTS 텍스트에 이어 붙인다.

- <audio> 엘리먼트

<audio> 엘리먼트를 만나면 'src' 속성 값을 추출하여 '|' 기호를 앞 뒤로 붙여 TTS 텍스트에 이어 붙인다.

- 텍스트 요소

텍스트 요소를 만나면 앞뒤 공백문자를 제거하고 TTS 텍스트에 이어 붙인다. 이와 같이 새롭게 생성된 TTS 텍스트는 Prompt 객체의 'script' 멤버 변수에 저장된다.

- <candidate> 엘리먼트

<prompt> 엘리먼트가 "alternative" 유형이어서 <candidate> 엘리먼트들을 포함한다면 그 하위의 요소들을 TTS 텍스트로 재구성하고 <candidate> 각각의 TTS 텍스트들을 Prompt 객체의 scriptList Vector 객체에 삽입한다.

6. 위자드-서브젝트 통신

위자드와 서브젝트 간의 통신은 TCP/IP 기반의 소켓 통신 방식을 이용한다. 전송 관리자는 시스템 간 메시지 수신을 위한 MessageTransfer 관리자와 오디오 데이터 송/수신을 위한 FileTransfer 관리자로 이루어져 있다.

1) MessageTransfer 관리자

MessageTransfer 관리자는 위자드와 서브젝트 사이에 파일 송/수신 및 작업 요청을 하는 역할을 한다. 메시지 송/수신을 위한 스레드를 생성한다. [표 4-3]과 [표 4-4]는 위자드와 서브젝트 시스템이 송/수신 하는 메시지와 메시지에 대한 설명이 나타나 있다.

[표 4-3] 위자드 시스템 송/수신 메시지

구분	메시지	설명
송신	START_TTS_PLAY	서브젝트 시스템에게 TTS 오디오 데이터 전송을 시작 할 것임을 알린다.
	STOP_AUD_REC	서브젝트 시스템에게 오디오 데이터 캡처 중지를 요청한다.
수신	SUCCESS_TTS_PLAY	서브젝트 시스템이 TTS 재생을 성공적으로 시작했음을 의미한다.
	END_TTS_PLAY	서브젝트 시스템이 TTS 재생을 종료했음을 의미한다. 위자드 시스템은 서브젝트로부터 전송되는 오디오 데이터 수신을 대기한다. 수신이 시작되면 실시간으로 전송된 오디오 데이터를 재생한다.
	END_AUD_REC	서브젝트 시스템이 오디오 캡처를 종료했음을 의미한다.

[표 4-4] 서버젝트 시스템 송/수신 메시지

구분	메시지	설명
송신	SUCCESS_TTS_PLAY	TTS 재생을 성공적으로 시작했음을 위자드 시스템에게 알린다.
	END_TTS_PLAY	위자드 시스템으로부터 수신된 프롬프트 재생이 완료되었을 때, 위자드 시스템에게 TTS 재생이 종료되었음을 알린다. 발화 데이터 캡처를 시작한다.
	END_AUD_REC	위자드 시스템으로부터의 캡처 종료 요청에 따라 서버젝트 시스템은 오디오 캡처를 종료하고 이를 위자드 시스템에게 알린다.
수신	START_TTS_PLAY	위자드 시스템으로부터 TTS 오디오 데이터 전송이 시작될 것임을 의미한다. 오디오 수신을 시작하고 전송된 데이터를 실시간으로 재생한다.
	STOP_AUD_REC	오디오 캡처를 중지한다.

2) FileTransfer 관리자

FileTransfer 관리자는 수신된 메시지 요청이나 위자드의 요청에 의해 오디오 데이터 송/수신을 시작하고 종료한다.

- SendThread

위자드-서브젝트 간에 오디오 데이터를 송신하기 위한 스레드이다.

- CaptureThread

SendThread 에 의해 전송된 오디오 데이터를 수집하기 위한 스레드이다. 수신된 오디오 데이터는 실시간으로 오디오 장치로 출력하여 재생한다.

- SaveThread




위자드 시스템에서만 생성된다. CaptureThread 가 종료되면 이 스레드가 생성되어 캡처된 오디오 데이터를 파일로 저장한다.

7. 시스템 수행 과정

시스템이 위자드/서브젝트 한 쌍으로 구성되지만 실제로는 위자드에 의해서 제어된다. 서브젝트 시스템은 위자드가 특정 작업을 요청할 경우에만 작업을 수행한다.

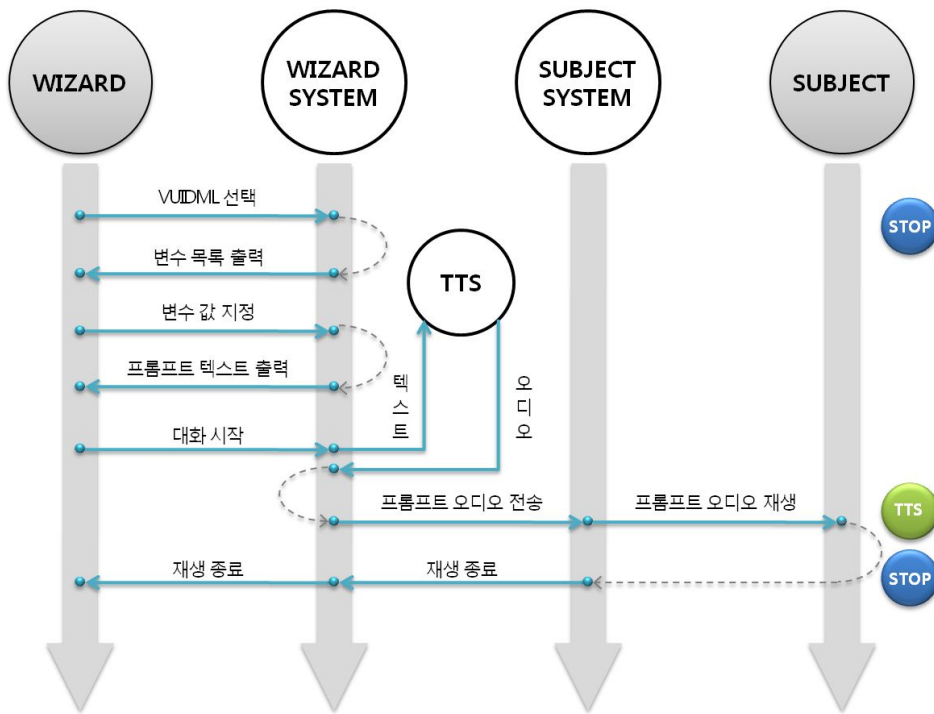
위자드에 의해 시스템이 수행하는 작업은 크게 ‘프롬프트 출력’, ‘발화 녹음’, ‘선호도 조사’가 있다. 이 세 가지 동작을 수행하는 과정에서 시스템은 [표 4-5]와 같은 세 개의 상태로 전환된다.

[표 4-5] 시스템 상태

기호	설명
	시스템이 대기 상태를 나타낸다. 프롬프트 재생이나 발화 녹음 상태를 제외한 경우이다.
	TTS를 통해 오디오로 변환된 프롬프트나 오디오 파일이 위자드 시스템에서 서브젝트 시스템으로 전송되어 재생되고 있는 상태를 나타낸다.
	위자드에 의해서 서브젝트 시스템이 사용자의 발화를 녹음하고 있는 상태를 나타낸다.

1) 프롬프트 출력

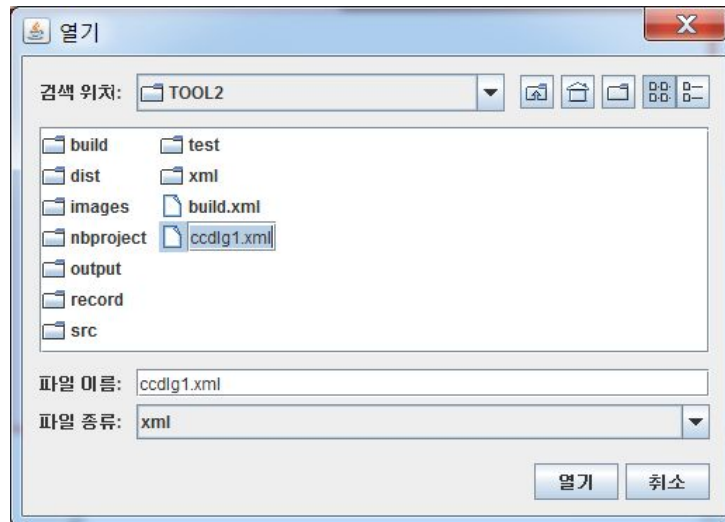
[그림 4-6]은 ‘프롬프트 출력’ 과정을 나타낸다. 이 과정이 종료되면 ‘발화 녹음’ 작업이 자동으로 시작된다.



[그림 4-6] 프롬프트 출력

◦ VUIDML 선택

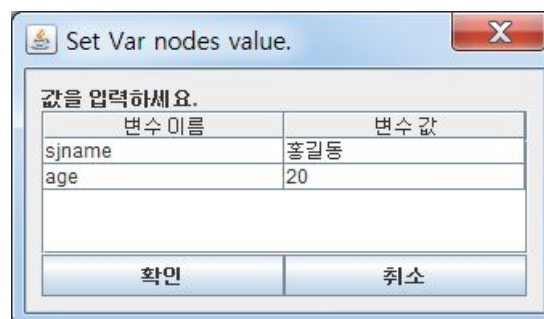
위자드는 시뮬레이션을 위한 VUIDML 문서를 선택한다. 시뮬레이션의 시작 시점에서만 필요한 작업이다. 파일을 여는 다이얼로그는 [그림 4-7]과 같다.



[그림 4-7] VUIDML 문서 선택 다이얼로그

◦ 변수 목록 출력

위자드 시스템에서는 VUIDML를 파싱하여 변수 목록을 사용자에게 출력한다. 시뮬레이션의 시작 시점에서만 필요한 작업이다. [그림 4-8]은 변수 목록이 출력되어 변수 값을 변경하는 다이얼로그를 보여 준다.



[그림 4-8] 변수 설정 다이얼로그

◦ 변수 값 지정

위자드가 변수 값을 새로 지정하면 VUIDML 문서에 입력한 값이 적용된다. VUIDML 문서를 선택하여 로드될 때 또는 시뮬레이션 도중에 위자

드 요청에 의해서 이 작업이 수행된다.

- 프롬프트 텍스트 출력

시스템은 변경된 VUIDML 문서의 프롬프트를 위자드에게 보여준다.

- 대화 시작

위자드가 대화 시뮬레이션을 시작하면 프롬프트 텍스트가 TTS 모듈을 통해 오디오 데이터로 변환된다.

- 프롬프트 오디오 전송

TTS 시스템이 변환한 오디오 데이터를 위자드 시스템으로 전송하면 위자드 시스템은 서브젝트 시스템으로 오디오 데이터를 전송한다.

- 프롬프트 오디오 재생

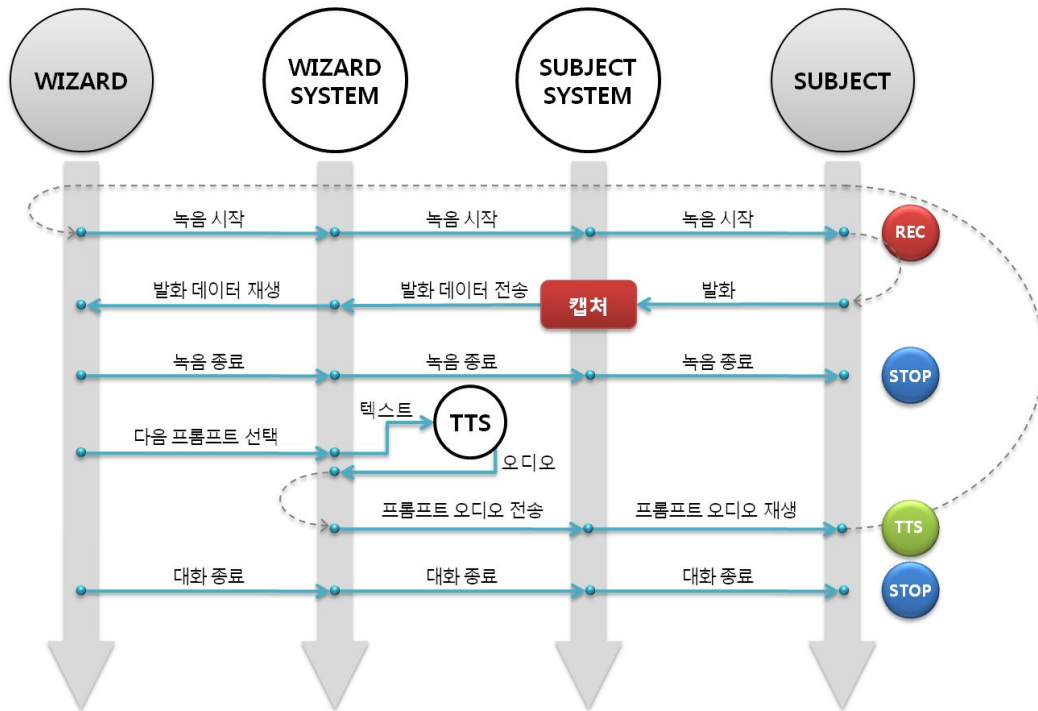
서브젝트 시스템은 전송된 오디오 데이터를 서브젝트에게 재생한다. 이 때 시스템은 'TTS' 상태로 전환된다.

- 재생 종료

재생이 완료되면 서브젝트 시스템은 위자드 시스템에게 재생이 종료되었음을 알린다. 이 때 시스템은 'STOP' 상태로 전환된다.

2) 발화 녹음

[그림 4-9]는 ‘발화 녹음’ 과정을 나타낸다. ‘발화 녹음’ 작업 후반부에는 ‘프롬프트 출력’ 작업이 수행된다. ‘프롬프트 출력’과 ‘발화 녹음’은 시뮬레이션 과정에서 반복적으로 수행된다.



[그림 4-9] 발화 녹음

◦ 녹음 시작

‘프롬프트 재생’이 종료되었다는 것을 위자드 시스템이 인지하면 녹음을 시작하는 요청을 서브젝트 시스템으로 전송한다. 서브젝트 시스템은 녹음 시작을 서브젝트에게 알린다. 이 때 시스템은 ‘REC’ 상태로 전환된다.

◦ 발화

‘REC’상태로 전환되면 서브젝트는 발화를 시작한다. 이 발화는 사용자

에게 주어진 작업을 수행하기 위해 시뮬레이션 시스템에게 내리는 명령을 의미한다.

- 발화 데이터 전송

‘REC’ 상태인 경우 서브젝트 시스템은 발화 내용을 오디오 데이터로 캡처하여 실시간으로 위자드 시스템에게 전송한다.

- 발화 데이터 재생

위자드 시스템은 전송된 오디오 데이터를 위자드가 들을 수 있도록 재생하여 위자드는 서브젝트의 발화 내용을 모니터링할 수 있다.

- 녹음 종료

서브젝트가 발화를 마치면 모니터 중인 위자드는 녹음을 종료하라는 요청을 서브젝트 시스템으로 전송한다. 서브젝트 시스템은 요청을 받아들여 캡처를 종료한다. 이 때 시스템은 ‘STOP’ 상태로 전환된다.

- 다음 프롬프트 선택

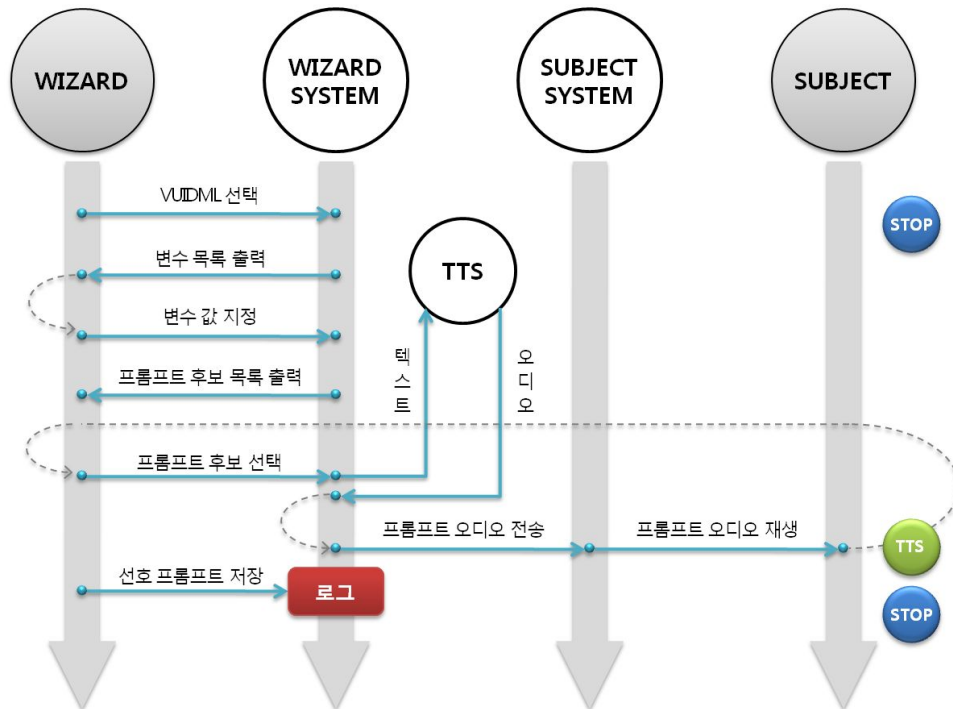
VUIDML에 설계된 시나리오에서 대화가 종료되지 않았다면 다음 프롬프트를 선택하여 ‘프롬프트 재생’을 시작한다. 다음 프롬프트는 위자드 시스템에 의해 자동으로 선택될 수도 있다.

- 대화 종료

VUIDML에 설계된 시나리오에서 대화가 완전히 종료되었으면 대화가 종료되었음을 서브젝트 시스템에게 알린다. 이 때 시스템은 ‘STOP’ 상태로 전환된다.

3) 선호도 조사

I. [그림 4-10]은 ‘선호도 조사’ 과정을 나타낸다. VUIDML 문서로 시물레이션을 진행할 때 ‘alternative’ 유형의 <block> 엘리먼트에 도달한 경우 서버젝트에게 프롬프트 후보들의 ‘선호도 조사’ 작업을 수행해야 한다.



[그림 4-10] 선호도 조사

- VUIDML 선택

‘프롬프트 출력’ 작업에서와 마찬가지로 위자드는 시물레이션을 위한 VUIDML 문서를 선택한다. 문서가 로드되어 있지 않는 경우에만 요구된다.

- ‘변수 목록 출력’, ‘변수 값 지정’

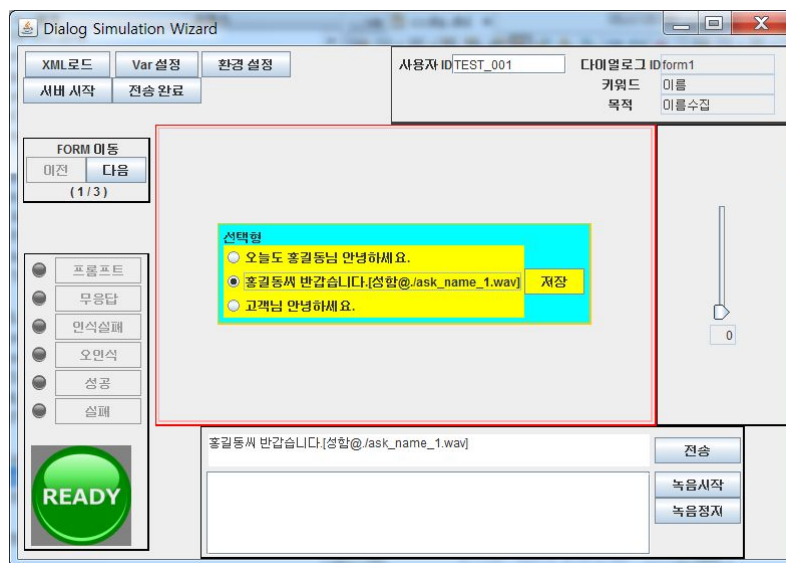
시물레이션을 시작하는 시점이 아닌 경우에는 건너뛴다.

- 프롬프트 후보 목록 출력

선호도 조사를 위해 VUIDML 문서 상에 설계한 프롬프트 후보 목록들이 위자드에게 출력된다.

- 프롬프트 후보 선택

위자드는 프롬프트 후보 목록 중에서 보이는 순서대로 선택해 서브젝트가 들을 수 있게 한다. [그림 4-11]은 프롬프트 선호도를 조사하기 위한 화면이다.



[그림 4-11] 프롬프트 선호도 조사 화면

- 프롬프트 오디오 전송

선택한 후보 프롬프트는 TTS에 의해 오디오 데이터로 변환되고 위자드 시스템은 서브젝트 시스템으로 오디오 데이터를 전송한다.

- 프롬프트 오디오 재생

위자드 시스템으로부터 전송이 시작되면 오디오 데이터를 재생한다. 재생이 완료되면 위자드는 다음 프롬프트 후보 목록을 출력한다.

◦ 선호 프롬프트 저장

위자드는 서브젝트에게 프롬프트 후보들을 각각 모두 들려주고 선호하는 프롬프트를 조사하여 응답 결과를 저장한다. 저장된 결과는 그림 [4-12]와 같은 구조의 XML 문서로 저장된다.

```
<choices>
  <choice promptID="ppt_1">
    <script index="1">
      흥길씨 반갑습니다.[성함@./ask_name_1.wav]
    </script>
  </choice>
</choices>
```

[그림 4-12] 선호 프롬프트 저장 구조

V. 결론

본 논문에서는 명령형 음성 사용자 인터페이스의 사용성 향상을 위해 실시하는 초기 사용성 평가를 자동화하기 위한 도구를 설계하고 구현하였다. 초기 사용성 평가는 위자드 오브 오즈 방식으로 실시된다는 가정하에 위자드 시스템과 서브젝트 시스템으로 각각 구성하였다.

설계한 사용성 평가 도구는 명령형 음성 인터페이스의 다이얼로그 시뮬레이션 시나리오가 요구된다. 이를 만족시키기 위해 기존에 개발되어있는 VoiceXML을 분석하여 수정하고 재구성하여 Voice User Interface Markup Language (VUIDML) 을 설계하였다.

본문에서는 설계한 VUIDML의 구성요소를 소개하였고, 이를 이용하여 작성한 시나리오의 예를 제시하고 시뮬레이션 되는 과정을 설명하였다. 또한 구현한 명령형 음성 사용자 인터페이스의 사용성 평가 도구의 구조와 동작 흐름을 기술하였다.

VUIDML과 사용성 평가 도구를 이용하여 음성 사용자 인터페이스의 사용성을 향상시키기 위한 명령어 수집과 사용자의 프롬프트의 선호도를 파악할 수 있다.

마지막으로 본 논문에서 제안한 VUIDML 문서를 자동으로 생성하고 문법을 검토해주는 기능을 갖는 VUIDML 설계 도구, 대화 로직을 설계하기 위한 명령형 사용성 평가 도구의 대화 흐름 시뮬레이션 기능, 초기 사용성 평가 단계의 결과 분석과 VoiceXML과 같은 완성된 VUI 설계 문서 생성 기능을 갖춘 도구가 향후 추가적으로 연구되어야 한다.

참고문헌 및 사이트

- [1] 김종훈, 심재호, 송창우, 이정현, “스마트 홈 환경에서 사용자 상황 정보 기반의 음성 인식 시스템 개발”, 한국콘텐츠학회논문지, 제8권, 제1호, 한국콘텐츠학회, pp. 328-338, 2008.
- [2] 김병창, 정민우, 이근배, “음성 인터페이스 기술의 현황과 전망”, 전자공학회지, 제34권, 제6호, 대한전자공학회, pp. 50-58, 2007.
- [3] 이윤근, 박준, “음성 인터페이스 기술”, 전자통신동향분석, 제20권, 제5호, 한국전자통신연구원, 2005.
- [4] 강윤환, 정성욱, 최재호, 정의승, “운전 상황에서 최적의 핸드폰 음성인식 기능 조사”, 대한인간공학회 2008 춘계 학술대회 논문집, 대한인간공학회, pp. 188-192, 2008.
- [5] 채행석, 홍지영, 전명훈, 한광희, “가전제품의 음성 인터페이스 디자인 적용에 대한 연구”, 감성과학, 제10권, 제1호, 한국감성과학회, pp. 55-68, 2007.
- [6] Jakob Nielsen, “Iterative User-Interface Design”, IEEE Computer, Vol. 26, No. 11, IEEE Computer Society Press, pp. 32-41, 1993.
- [7] 홍기형, “음성기반 멀티모달 사용자 인터페이스의 사용성 평가 방법론”, 말소리, 제67호, 대한음성학회, pp. 103-120, 2008.
- [8] Bob Edgar, “The Voice XML Handbook”, CMP, 2001.
- [9] W3C, “Voice eXtensible Markup Language (VoiceXML) version 1.0”, <http://www.w3.org/TR/voicexml>, 2000.
- [10] W3C, “Voice Extensible Markup Language (VoiceXML) 2.1”,

<http://www.w3.org/TR/voicexml21>, 2007.

- [11] Carol M. Barnum, Sam Dragg, “Usability Testing and Research, Longman”, 2001.
- [12] 조셉 두마스, 제니스 레디쉬, “사용성 테스트 가이드 북”, 한솜미디어, 2004.
- [13] Jakob Nielsen, “Usability 101: Introduction to Usability, <http://www.useit.com/alertbox/20030825.html>.
- [14] Randolph G. Bias, Deborah J. Mayhew, “Cost-justifying usability”, Academic Pr, 2005.
- [15] Jonh D. Gould, Clayton Lewis, “Designing for usability: Key Principles and What Designers Think”, Communications of the ACM, Vol. 28, No. 3, ACM, pp. 300-311, 1985.
- [16] 한우석, “사용성 평가 (Usability Testing) 조사 소개”, 리서치노트, 봄호, pp. 36 - 41, 2009.
- [17] Saul Greenberg, “Prototyping for Design and Evaluation”, <http://grouplab.cpsc.ucalgary.ca/saul/681/1998/prototyping/survey.html>, 1998.
- [18] 권지혜, 홍기형, “음성 사용자 인터페이스 설계 방법론”, 정보과학회지, 제24권, 제1호, 한국정보과학회, pp. 27-36, 2006.
- [19] Michael H. Cohen, James P. Giangola, Jennifer Balogh, Voice User Interface Design, Addison-Wesley, 2004.
- [20] 한성호, 김범수, “음성인식용 인터페이스의 사용편의성 시험 방법론”, 대한인간공학회지, 제18권, 제3호, 대한인간공학회, pp. 105-125, 1999.

- [21] UPA(Usability Professionals' Association), "What is User-Centered Design", http://www.upassoc.org/usability_resources/about_usability/what_is_ucd.html.
- [22] Carol Smith, Justin Thorp, Shawn Lawton Henry, "Note on User Centered Design Process(UCD)", <http://www.w3.org/WAI/EO/2003/ucd>, 2004.
- [23] Kyle Soucy, "Unmoderated, Remote Usability Testing: Good or Evil?", <http://www.uxmatters.com/mt/archives/2010/01/unmoderated-remote-usability-testing-good-or-evil.php>, 2010.
- [24] 오기태, 이건표, "웹 사이트 원격 사용성 테스트에 관한 연구: 원격 사용자 인터랙션 관찰 및 분석 도구의 개발을 중심으로", 디자인학연구, 제24권, 제3호, 한국디자인학회, pp. 147-156, 2004.
- [25] Craig Tomlin, "24 Usability Testing Tools", <http://www.usefulusability.com/24-usability-testing-tools>, 2009.
- [26] Scott R. Klemmer, Anoop K. Sinha, Jack Chen, James A. Landay, Nadeem Aboobaker, Annie Wang, "Suede: a Wizard of Oz prototyping tool for speech user interfaces", UIST '00: Proceedings of the 13th annual ACM symposium on User interface software and technology, ACM, 2000.

ABSTRACT

Design and Implementation of A Usability Testing Tool for Command-and-Control Voice User Interfaces

Myeongji Lee

Department of Computer Science

Graduate School of Sungshin Women's University

Recently, usability becomes very important in voice user interface systems. In this thesis, we have designed and implemented a wizard-of-oz(WOZ) usability testing tool for command-and-control voice user interfaces. We have proposed the VUIDML (Voice User Interface Design Markup Language) to design the usability test scenario of command-and-control voice interfaces in the early design stages. For the highly satisfied voice user interfaces, we have to select highly preferred voice commands and prompts. In VUIDML, we can specify possible prompt candidates. The WOZ usability testing tool can be used to collect user-preferred voice commands and feedbacks from real users.