

# MS Agent의 명령어 생성기의 설계

김장우\*, 최영미\*\*

\*성결대학교 컴퓨터학부

\*\*성결대학교 멀티미디어학부

## A Study on Design of MS Agent Command Generator

Jangwoo Kim\*, Youngmee Choi\*\*

\* Division of Computer, Sungkyul University

\*\* Division of Multimedia, Sungkyul University

### 요 약

본 연구에서는 마이크로소프트 에이전트의 보다 효율적인 활용을 위하여 GUI 방식으로 마이크로소프트 에이전트 스크립트 생성기를 설계하고 구현하였다. 적용사례에서 기존의 스크립트 코딩 방식과 본 연구에서 고안한 GUI 방식으로 각각 제작과정을 분석하여 효율성을 보였다. 또한 개선사항으로 다국어 사용의 편의성과 캐릭터 크기 조정 기능은 마이크로소프트 에이전트의 사용성을 더욱 높여 주었다.

### 1. 서론

인간과 컴퓨터의 인터페이스 기술은 컴퓨터 기술의 발전과 사용자의 요구에 따라 진화해 오고 있다. 과거 도스 환경에서 사용자가 콘솔 화면에 직접 명령어를 타이핑하여 입력하는 방식이었으나, 윈도우 환경에서는 사용자가 필요한 명령어 아이콘을 클릭하는 방식으로 컴퓨터와의 상호작용을 매우 쉽고 편리하게 해 주었다. 최근에는 에이전트 기술의 등장으로 다양한 방식의 상호작용 기술이 시도되고 있다. 이러한 기술들 중 캐릭터 에이전트는 인간과 컴퓨터의 매개체로 캐릭터를 사용하여 인터페이스에서 사용자의 비서와 같은 존재로서 사용자를 대신하여 복잡한 일을 처리하는 대리자를 의미한다. GUI에 캐릭터 에이전트의 등장은 인터페이스의 접근방식의 변화를 예고하고 있다. 즉 캐릭터를 인터페이스에 등장시켜 보다 자연스럽게 인간적이며 직관적인 상호작용을 제시한다.

그러나 마이크로소프트 에이전트를 프레젠테이션에 구현시 사용자가 직접 스크립트를 코딩해야하는 것은 프레젠테이션을 준비하는 사용자에게는 또 다른 부담이 되어 적극적으로 활용되지 못하고 있다. 본 연구에서는 이러한 문제점을 해결하기 위하여

GUI환경에서 에이전트 스크립트를 자동으로 생성하는 인터페이스를 설계하고 구현하였다.

본 논문에서는 에이전트 명령어 생성기(ACG: Agent Command Generator) 인터페이스를 설계하고 구현하였다. 파워포인트 강의문서에 마이크로소프트 에이전트를 구현하는 사례를 ACG를 사용하여 제작하여 기존 제작방식과 비교분석을 통하여 ACG의 효율성과 개선사항을 보인다. 2장에서 마이크로소프트 에이전트의 개념과 특성을 고찰하고 기존의 명령어 코딩 방식의 문제점을 도출하고, 3장에서 그 해결방안으로 GUI 방식의 에이전트 명령어 생성기를 설계하고 구현하였다. 4장에서 적용사례로 ACG를 사용하여 에이전트 기반 프레젠테이션을 구현한 후, 기존 방식과 비교하여 각각 제작과정을 분석하고 개선된 사항을 논의한다. 마지막으로 5장에서는 마이크로소프트 에이전트의 응용분야와 앞으로의 연구 방향을 제시하면서 결론을 맺는다.

### 2. 마이크로소프트 에이전트

마이크로소프트 에이전트는 하나의 소프트웨어 기술로써 컴퓨터의 사용과 학습을 좀 더 쉽고 자연스럽게 만들어 줄 수 있다. 또한 개발자는 쉽게 자신의

응용 프로그램이나 웹사이트에 대화도 가능한 살아 있는 듯한 캐릭터를 삽입하여 사용자 인터페이스를 향상시킬 수 있다. 사용자는 마이크로소프트 에이전트가 지원하는 대화형 인터페이스 접근을 통해 마이크로소프트 윈도우 환경 하에서 강력한 확장과 개선된 모습을 보여줄 수 있다.

- MS Agent 는 소프트웨어를 배우는 것을 쉽게 해 주고 능력을 향상시켜준다
- MS Agent를 설치하면 개발자들은 그들의 어플리케이션과 웹사이트의 인터페이스의 움직임을 향상시켜 준다.
- 캐릭터들은 자유롭게 말하고 움직일 수 있다.

즉, 마이크로소프트 에이전트의 자유롭게 움직일 수 있고, 소리를 내어 말하고, 풍선도움말 형태로 문자 표현을 하고, 음성인식 기능들의 융합은 사용자에게 가능한 대화형 양식들을 경험할 수 있게 한다. 이때, 캐릭터는 사용자 인터페이스에서 사용자의 '비서'와 같은 존재로서 사용자를 대신하여 복잡한 일을 처리하는 대리자를 의미한다. 이러한 대화를 지원하는 에이전트를 사용하면 사용자의 상호대화의 영역도 확장되며 강력한 힘을 발휘할 수 있게 된다 [5-7].

## 2.1 마이크로소프트 에이전트의 특징

마이크로소프트 에이전트의 특징은 다음과 같다. 첫째, 어플리케이션과 웹사이트의 사용자 인터페이스를 확장하고 향상시킨다. 애니메이션은 윈도우 어디서나 움직임이 가능하고 윈도우와 분리되어 자유롭다. 엔진은 문자를 언어로 바꾸어주며 오디오에 녹음도 된다. 임의의 어떤 단어는 엔진에 명령어로 입력이 된다. 둘째, 포괄적인 프로그래밍을 지원한다. 모든 프로그래밍언어, HTML, 스크립트 언어를 통한 웹페이지에서도 호환이 가능하며 애니메이션, 언어, 명령인식, 이벤트의 동기화로 보다 넓은 제어가 가능하다. 셋째, 개발 요구를 융통성 있고 확장성 있게 지원한다. 비록 낮은 모델을 사용하더라도 낮은 런타임 대역폭을 요구하므로 캐릭터의 애니메이션을 계속하면서 무리 없이 실시간 컴파일해 준다. 외국어 엔진을 통해 유연성있게 외국어로 대화하는 것도 가능하다. 넷째, MS Office와 다른 응용에 에이전트를 통합시킬 수 있고, ActiveX 기술, VBscript, JScript 언어

를 사용하여 웹페이지, 파워포인트 등에 다양한 마이크로소프트 에이전트를 구현 시킬 수 있다[1-4].

## 2.2 마이크로소프트 동작 요소 분석

본 논문에서 동작은 마이크로소프트 에이전트를 파워포인트에 구현하는 명령어를 의미한다. 여기서 핵심적인 명령어를 소개하면 SAY(말하기), PLAY(애니메이션), MOVE(이동), SIZE(크기조절)이다(그림 1).



그림 1. 동작 요소

PLAY는 캐릭터가 개발자가 지정한 행동을 실행하도록 하는 역할을 한다. SAY는 캐릭터가 이 명령어 뒤에 오는 문장을 읽도록 하는 역할을 한다. MOVE는 이 명령어에서 기술한 좌표 값의 위치로 캐릭터가 이동하도록 하는 역할을 한다. SIZE는 캐릭터의 크기를 조절한다. SIZE는 본 연구에서 추가한 기능이다. 그 외에 마이크로소프트 에이전트에서 제공하는 명령어들은

SAYNOCAP(글없이 말하기), AUDIO(소리), AUDIONOCAP(글없는 소리), THINK(생각), LOOP(반복), SLIDE(다음 슬라이드로 가라), SHOW(등장), HIDE(퇴장), SUSPEND(쉬기), WAIT(기다림), END(종료), COMMENT(주석), RUN(실행), NEXT(다음), SOUNDFXON(효과음켜기), SOUNDFXOFF(효과음끄기) 등이 있다.

예를 들어 “캐릭터가 오른쪽을 가리키며 ‘I can act as a sales representative or lead you through a series of questions to arrive at a final decision’라고 말하고 프레젠테이션 화면의 좌표(30,20)로 이동하면서 크기가 2배로 커진다.”를 표현하는 동작요소를 명령어 리스트로 표현하면 그림 2과 같다.

```

PLAY GestureRight
SAY I can act as a sales representative or lead you
through a series of questions to arrive at a final decision.
MOVE 30,20
SIZE 200%
  
```

그림 2. 마이크로소프트 명령어 리스트

### 2.3 기존 마이크로소프트 에이전트 명령어 생성의 문제점

개발자가 마이크로소프트 에이전트 명령어를 작성 시 실제 환경에서 애니메이션 동작모습을 확인하기 어렵고 정확한 위치를 선정하려면 반복되는 수정작업이 필요하고, 다국어 사용시 내부코드를 수정해야 하는 등 개발자들에게 부담을 주고 있다. 이러한 문제점을 정리하면 표 2와 같다.

표 1. 기존 개발 방식의 비효율성

기능요소	내용
애니메이션	text 상의 애니메이션 리스트만으로는 해당 캐릭터가 실제로 동작하는 모습을 확인하기 어렵다.
위치이동	캐릭터가 화면에서 위치이동을 하는데 감으로 대략의 위치를 정하여 움직일 수는 있으나 파워포인트의 특정 위치에서 애니메이션 하기위해서는 실제 동작하는 화면에서 확인이 필요하고 적당하지 않을 경우에는 수정하고 확인하는 작업을 계속 반복해야 한다.
기타	이 MS Agent는 다양한 언어를 사용하며 그에 알맞은 언어로 말을 할 수 있다. 그러나 기본적으로 제공되는 코드 샘플에는 한 캐릭터가 하나의 언어만을 사용하는 것밖에 제공되지 않는다. 그러나 동시에 여러 언어를 사용할 수 있다는 가능성은 시사하고 있으나 프로그래밍 지식이 없는 사람들에게는 이 기능을 충분히 사용하기에는 무리가 따른다.

### 3.1 ACG 명령어 리스트 생성

에이전트 기반 프레젠테이션 구조의 작동원리는 다음과 같다. 사용자가 주제를 선정하여 파워 포인트 자료를 작성한 후 이에 기초하여 동작 시나리오를 생성한다.

생성된 시나리오를 기반으로 그림 3과 같은 단계를 거쳐 최종 완성된 명령어 리스트를 작성한다. 세부 내용은 다음과 같다.

- ① 캐릭터 선택 : 준비되어 있는 4가지 캐릭터 중 어울리는 캐릭터를 선택한다.
- ② 선택한 캐릭터 등장 : 캐릭터가 화면에 나타나면서 여러 가지 내용이 수정된다.
- ③ 동작요소 선택 : 시나리오에 알맞은 요소들을 선택한다.
- ④ 선택된 동작요소 프리뷰 : 캐릭터가 선택된 동작요소를 보여준다.
- ⑤ 동작요소 추가/삽입 : 선택된 동작요소를 명령어 리스트에 추가/삽입을 한다.
- ⑥ 명령어 리스트 프리뷰 : 명령어 리스트의 동작요소를 프리뷰 한다.
- ⑦ 동작요소 편집(수정/삭제) : 명령어 리스트의 동작요소를 편집한다.
- ⑧ 최종 완성 : 위의 모든 단계를 반복하여 완료되면 실제 파워포인트의 메모장에 붙여 넣을 수 있는 상태로 만든다.

### 3. Agent 기반 프레젠테이션 구현

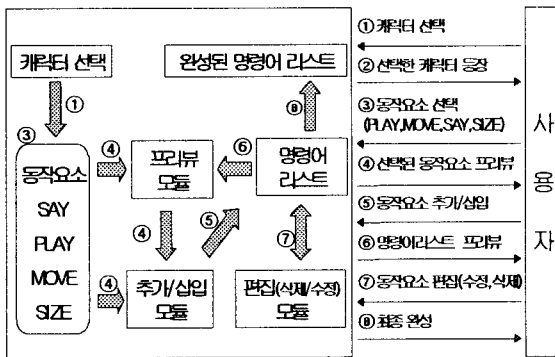


그림 3. ACG 구조

본 연구에서 구현한 ACG는 프레젠테이션에 에이전트 구현을 효율적으로 할 수 있도록 명령어 생성 및 편집 프리뷰를 제공한다. ACG에서 작성/수정/보완된 페이지 단위의 명령어 리스트는 파워포인트의 메모장에 복사하여 애니메이션 캐릭터가 구현된 프레젠테이션이 제작된다. ACG의 인터페이스 구조는 그림 3과 같다.

프레젠테이션 내용과 동작 시나리오에 알맞은 캐릭터를 캐릭터 라이브러리에서 선택하여 적절한 속성을 사용하여 에이전트 명령어를 생성한다.

### 3.2 ACG의 인터페이스 구조

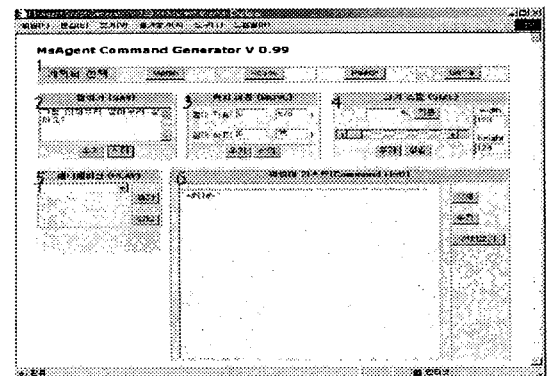
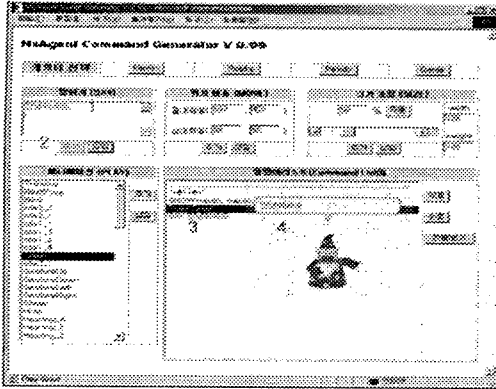


그림 4. ACG 사용자 인터페이스

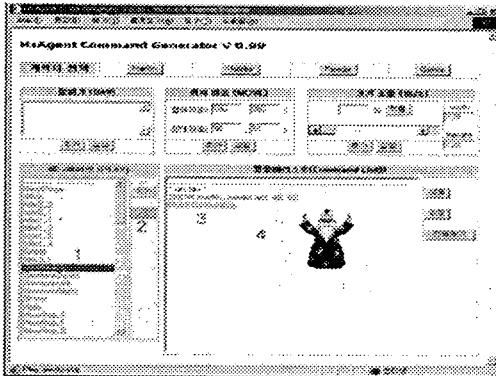
그림 4와 같이 ACG 사용자 인터페이스 구조는 6개의 요소와 최종 결과물 1개로 구성되어 있다. 각 명령어의 생성 및 편집을 구분하여 기술하면 다음과 같다.

(1) 말하기(SAY)



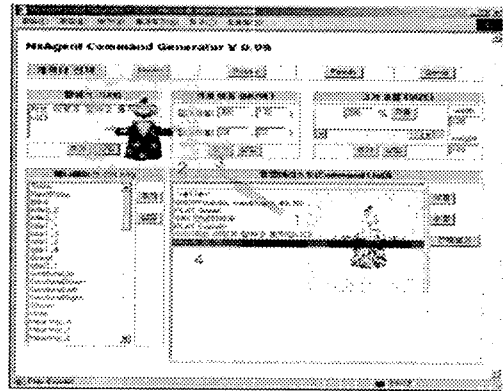
말하기 영역에서 1번과 같이 문장을 입력하고 엔터를 누르면 4번과 같이 캐릭터는 입력한 문장을 풍선에 표시하고 문장을 읽어준다.  
(참고로 동작은 애니메이션 추가 후 변화를 주지 않았기 때문에 그대로 있다. 그러나 어떤 애니메이션은 동작을 완료하면 RestPose로 되돌아간다.)  
2번의 추가 버튼을 누르면 명령어 리스트에 애니메이션을 추가한 것처럼 파워포인트에서 사용하는 방식의 명령어가 생성된다.

(2) 애니메이션(PLAY)



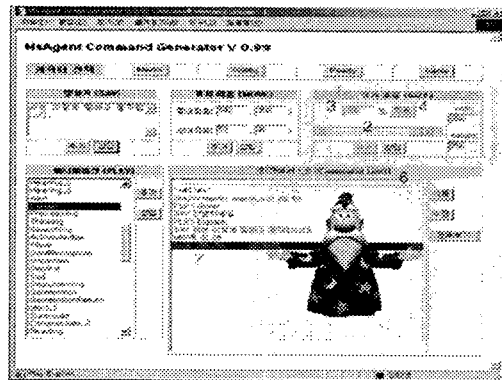
1번과 같이 애니메이션 영역에서 Greet을 선택하면 4번과 같이 캐릭터가 해당하는 동작을 한다.  
그 다음 2번의 추가 버튼을 누르면 3번처럼 명령어 리스트에 추가된다.

(3) 이동(MOVE)



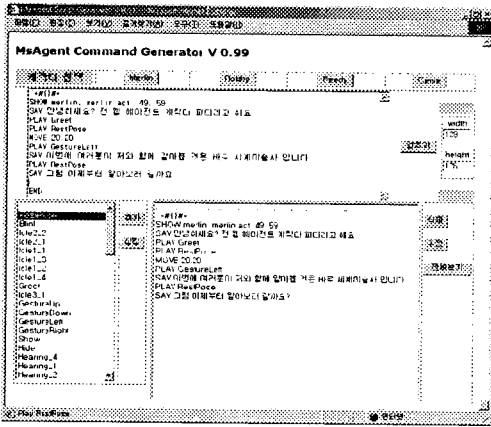
캐릭터를 1번 위치에서 2번 위치로 이동하면 위치 정보에서 각 좌표들이 바뀌게 된다.  
이때 3번의 이동 추가 버튼을 누르면 4번과 같이 명령어 리스트에 해당 명령어가 이동 위치 좌표와 함께 추가된다.(1.캐릭터의 위치는 사용자의 화면 크기의 비율로 표현되기 때문 제작당시의 화면 해상도와 시연을 할 때의 화면 해상도와 일치시켜주는 것이 바람직하다, 2.캐릭터의 위치는 전체화면 크기를 벗어날 수 없도록 위치 이동 제한 기능을 추가하였다.)

(4) 크기 조절(SIZE)



1번의 크기 조절 영역에서 2번의 슬라이드 바를 오른쪽 끝으로 이동하면 3번의 배율 숫자가 최대 200%까지 변한다. 이때 4번의 적용 버튼을 눌러주면 6번과 같이 캐릭터의 사이즈가 원래 사이즈의 200%, 즉 두 배로 커진다. (실제 더 커질 수 있으나 200%로 제한했다.)  
사이즈를 확인하고 5번의 추가 버튼을 누르면 7번과 같이 명령어 리스트에 추가된다.

(5) 생성된 명령어 리스트 편집



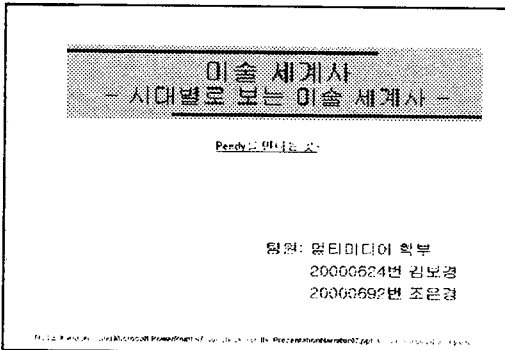
삭제, 수정의 작업은 (1)~(4)의 과정이 반복되며 각 위치의 버튼(삽입, 삭제, 수정)을 이용하여 생성된 명령어 리스트를 편집할 수 있다. 모든 단계가 완료되면 전체보기 버튼을 눌러 최종 결과를 화면을 보이게 한 다음 복사하여 파워포인트 메모장에 붙인 후 시연을 한다.

4. 적용사례 및 결과 분석

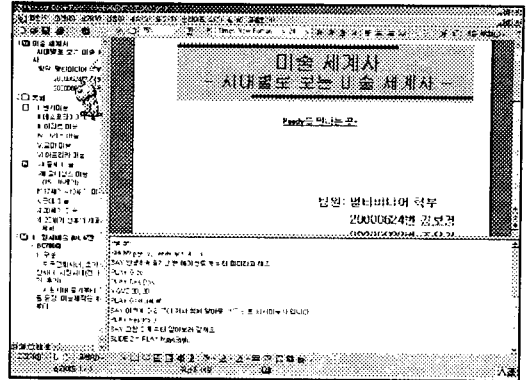
4.1 적용사례

강의 주제는 마이크로소프트 에이전트를 소개하는 내용으로 캐릭터는 프레젠테이션에서 교수의 역할에 알맞은 Merlin을 선택하였다. 적용 과정은 다음과 같다.

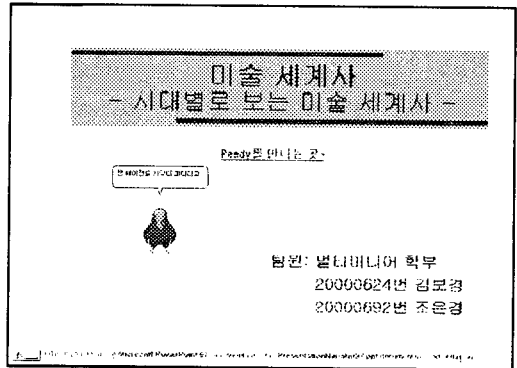
(1) 파워포인트를 작성한다.



(2) ACG로 작성한 코드를 파워포인트의 메모장에 명령어 리스트를 붙인다.



(3) 시연모습



4.2 결과 분석

위의 적용사례를 학생들을 통해 비교 평가해본 결과는 표 2와 같다.

표 2. 비교 평가 및 결과

비교 항목	기존 개발 방법	ACG를 이용한 개발 방법
MOVE	1. 캐릭터의 위치 선정 2. 대략 위치를 상대좌표로 생각해서 입력 3. 시연을 하여 위치를 찾을 때 까지 2~3번을 수 회 반복	1. 캐릭터의 위치 선정 2. 시연을 한다. 3. 캐릭터를 선정된 위치로 이동 4. 명령어 리스트에 추가
	소요 시간 : 수 분 ~ 십 수분	소요 시간 : 1분 이내
PLAY	1. 시나리오에 맞는 애니메이션 선택 2. 시연을 하여 선택된 애니메이션 확인 3. 아닐 경우 재확인(수회 반복)	1. 시나리오에 맞는 애니메이션 선택 2. 애니메이션 메뉴에서 동작 확인 3. 명령어 리스트에 추가
	소요 시간 : 수 분 ~ 십 수분	소요 시간 : 1분 이내
SAY	하나의 언어만을 사용함	다국어 사용 가능
SIZE	기본 크기만 사용	1%~200%(그 이상 가능) 확대/축소 가능

## 5. 결론

ACG를 이용하여 위의 결과 분석의 내용처럼 개발을 하는데 효율적이며 확장된 기능을 손쉽게 구현할 수 있게 되었다.

본 연구에서 개선된 사항은 다국어 사용의 편의성(intelligent SAY)과 캐릭터 크기 조정(SIZE)이 가능한 점이다.

특정 언어를 사용하기 위해 별도의 명령을 사용하지 않고 SAY의 내용을 분석하여 해당 언어로 자동 전환 시키며 현재 한글과 영어의 혼용이 가능하다.

기존에는 정해진 크기만을 사용하였는데 캐릭터의 크기를 조절 할 수 있는 융통성을 부가함으로써 보다 효과적이고, 확장된 인터페이스를 제공한다.

본 연구의 응용 가능한 분야는 시각 장애인을 위한 홈페이지 제작, 실버산업(쇼핑몰), TTS(Text To Speech)를 이용한 음성채팅, E-MAIL Reader, 아동 학습 사이트 등 기존 읽기 위주의 환경들을 상호 작용성을 높이고 음성 출력을 필요로 하는 사용자들을 위한 사이트를 제작하는데 있다.

앞으로 연구과제는 자연스러운 음성 출력과 동적인 ACG 개발에 있다. 영어 기반의 TTS를 사용하여 한글의 표현이 부자연스럽다. MS에서 제공하는 TTS는 기존 한글 전용 TTS보다 용량이 작아 인터넷 환경에서 쉽게 사용할 수 있다. 그러나 한글 표현의 부자연스러움을 극복하기 위한 방안으로 스피드, 피치, 볼륨 등을 조절하여 보다 자연스러운 음성 출력을 위한 연구가 필요하다. 파워포인트 같은 어플리케이션에서 MS Agent를 사용하면 상당히 정적인 프래젠테이션이 되는데 반해 인터넷에서 이러한 제약 없이 줄일 수 있다. 인터넷상에서 상호작용성 높고 동적인 환경에서 적절한 ACG의 연구를 앞으로의 과제로 남긴다.

conversational agrnts", Readings in agents pp148-155, 1998

[7] Brian A. Stone and James C. Lester, "Dynamically Sequencing an Animated Pedagogical Agent", Readings in agents, pp156-163, 1998

## [참고문헌]

- [1] <http://www.extempo.com>
- [2] <http://www.microsoft.com/msagent>
- [3] <http://www.msagentring.org>
- [4] <http://www.bellcraft.com/mash>
- [5] Arthur C. Graesser, AutoTutor: A Simulation of a Human Tutor, Journal of Cognitive Systems Research.
- [6] Justine Cassell, "Animated conversation: Rule based generation of facial expression, gesture & spoken intonation for multiple