

응답 관계의 효율적인 프레젠테이션을 지원하는 트리 기반 대화 인터페이스

김 경 덕[†]

요 약

본 논문에서는 온라인 대화에서 대화자간 대화 메시지의 응답 관계를 효율적으로 프레젠테이션 하는 트리 기반 대화 인터페이스를 제안한다. 기존의 대화 인터페이스는 대화 인터페이스의 단순성과 대화 메시지 자료 구조의 제약으로 인하여 그룹 토의나 의사 결정과 같은 형식적인 대화를 지원하기 어려우며, 트리 기반 대화 인터페이스는 형식적인 대화의 지원은 가능하나 응답 관계에 의한 대화창의 이동 및 응답간의 거리에 의하여 대화 메시지의 프레젠테이션 등이 어렵다. 그러므로 본 논문에서는 텍스트 기반 대화 인터페이스의 장점과 XML 기반 메시지를 이용하여 대화에서 응답 관계 프레젠테이션을 효율적으로 지원하는 트리 기반 대화 인터페이스를 기술한다. 제안한 대화 인터페이스의 구현은 XML, DOM, JDK를 이용하여 클라이언트/서버 구조의 대화 시스템에 적용한 예를 보였으며, 응용 분야는 협업, 원격 교육, 온라인 게임 등이다.

Tree-Based Conversational Interface Supporting Efficient Presentation of Turn Relations

Kyungdeok Kim[†]

ABSTRACT

This paper describes a tree-based conversational interface supporting efficient presentation of turn relations on online conversation. Most of conventional conversational interfaces are difficult to make use of formal conversation such as group meeting, decision-making, etc. due to very simplicity of a conversational interface and restriction of data structure of conversational messages. And a tree-based conversational interface supports formal conversation, but they are difficult to present turn relations because of jumpy display by locations of replied turns and distance between replied turns, etc. So this paper suggests a tree-based conversational interface to present efficiently turn relations using XML-based messages with merits of a text-based interface. The suggested conversational interface was implemented by using XML, DOM, and JDK. And this paper showed that the conversational interface could be applied to conversation system using client-server architecture. Applications for the conversational interface are as follows: collaboration, distance learning, online game, etc.

Key words: Conversational Interface(대화 인터페이스), Online Conversation(온라인 대화), Turn Relation(응답 관계), Conversational Messages(대화 메시지), and Formal Conversation(형식 대화)

※ 교신저자(Corresponding Author) : 김경덕, 주소 : 경북 경주시 강동면 유급리 산 50(780-713), 전화 : 054)760-1655, FAX : 054)760-1719, E-mail : kdkim@mail.uiduk.ac.kr
접수일 : 2003년 7월 4일, 완료일 : 2003년 8월 12일
[†] 정회원, 위덕대학교 컴퓨터멀티미디어공학부 조교수

1. 서 론

온라인 대화는 컴퓨터 통신 기술의 발달과 인터넷의 확장으로 교육 및 게임 등에서 활용이 증가하고

있으며, 또한 이동 통신과 연동하여 다양한 서비스 기술들이 개발되고 있다[1-4,10]. 그러나 아직 온라인 대화를 지원하기 위한 대화 메시지의 자료 구조나 대화 인터페이스는 기존 텍스트 기반의 대화 인터페이스 형식에서 벗어나지 못하고 있다. 이러한 이유로 기존 대화 시스템들은 오락 및 교육 등을 위한 비형식 대화(informal conversation)에는 범용적으로 이용되고 있지만, 사무 처리를 위한 의사 결정과 같은 형식 대화(formal conversation)에서는 사용이 어렵다[4-6]. 현재 다양한 대화 시스템이 출시되고 있으나, 대부분의 대화 인터페이스는 기존의 대화 메시지 및 단순 텍스트기반 대화 인터페이스를 응용한 형식을 이용하고 있으며, 2D 및 3D의 그래픽 표현을 적용하여 대화에서 부분적인 사회 정보(social information)를 표현하고 있는 정도이다[11,13].

일반적인 오프라인 대화에서는 다양한 주제를 중심으로 자유로운 대화가 이루어지며, 대화 중에 대화의 중단과 다자간 대화시 대화의 중첩, 그리고 대화 중에 대화자의 변경이 이루어질 때, 대화의 진행에 지연이 없이 부드럽게 다른 대화로 이어진다[4]. 그런 반면, 온라인 대화는 컴퓨터에 의하여 중재되는 실시간 통신의 형태로서 오프라인 대화와 유사하지만, 오프라인 대화에서 이루어지는 대화에서 끼어들기, 대화자간 응답 관계의 체계화, 대화의 의미 전달, 대화자의 발언권 제어를 온라인 대화에서 지원하기는 매우 어렵다[4-6]. 이러한 온라인 대화에서 발생하는 대화 정보의 모호함과 대화 흐름의 변화 등의 여러 가지 문제점들을 해결하기 위하여 사회학, 통신, CSCW(Computer Supported Cooperative Work), HCI, 언어학, 인간공학 등 다양한 분야에서 많은 연구가 진행되고 있다.

일반적인 온라인 대화에서 이용하는 대부분의 대화 인터페이스는 주로 텍스트 기반 인터페이스나 트리 기반 인터페이스를 응용하고 있다. 기존 텍스트 및 트리 기반 대화 인터페이스의 특징과 문제점을 살펴보면 다음과 같다. 먼저, 기존 텍스트 기반 대화 인터페이스는 대화자가 자신의 의사를 키보드 등을 이용하여 편리하게 기술할 수 있고, 사용 방법의 습득이 용이한 대화 인터페이스로서, 주요 문제점은 다음과 같다[4-7]. 첫째, 대화 메시지는 대화자의 구별이 어렵다. 특히, 대화 메시지의 발생이 많으면 많을수록 대화자의 구별은 더욱 어려우며, 이를 부분적

으로 해결하기 위하여 대화자별로 대화 메시지에 색깔, 서체, 이미지 아이콘 등을 지정하여 사용하지만 대화자의 구별은 여전히 미흡하다. 둘째, 대화 참여자의 각 대화 메시지에 따라 누가 자신의 대화 메시지에 응답하고 있는지에 대한 정보를 파악하기 어려워 대화 내용의 전반적인 이해가 어렵다. 셋째, 대화 메시지의 작성 과정을 시각화하기 어렵다. 이런 시각화의 어려움은 대화 시스템이 대화 메시지가 작성된 후에 전달되기 때문이다. 즉, 대화 메시지의 생성과정과 전달 과정이 구분됨으로 발생한다. 넷째, 각 대화 메시지에 해당하는 응답 메시지가 연관되도록 위치시키기 어렵다. 각 대화 메시지의 응답 관계 모호성으로 인하여 대화자는 이전의 대화 메시지들을 탐색하여 그 관계를 유추해야 함으로 대화 내용의 파악이 어렵다. 이러한 응답 관계의 모호성은 대화 메시지가 대화 서버를 통하여 전달되는 과정에서 대화 메시지의 정렬이 대화 메시지의 작성 순서가 아닌, 대화 서버에 대화 메시지가 수신되는 순서로 상대방 대화자의 대화 인터페이스에 정렬됨으로써 발생한다. 다섯째, 대화 메시지의 내용과 연관성을 지속적으로 관리하기 어렵다. 대부분의 대화 인터페이스가 대화자에게 전체 대화 메시지 중에 일부분만 제공하고 대화 내용의 저장과 관리를 지원하지 않으며, 지원하더라도 대화 내용의 모호성으로 인하여 대화 내용의 정렬 등이 어렵다. 그리고 트리 기반 대화 인터페이스는 트리 구조에 기반을 두어 대화자가 작성하는 대화 메시지의 응답 관계를 계층적으로 나타내는 대화 인터페이스로서, 주요 문제점은 다음과 같다[4-6,14]. 첫째, 최근 수신한 대화 메시지의 파악이 어렵다. 즉, 서로 연관성이 있는 대화 메시지들의 수신은 응답 관계로 인하여 공간적으로 가까운 곳에 위치하고 대화창의 이동이 거의 없으나, 서로 연관성이 없는 대화 메시지는 지역적으로 떨어진 곳에 위치하게 됨으로써, 최근에 어떤 대화자가 무슨 대화 메시지를 작성하였는지를 파악하기 위하여 전체 대화 메시지를 탐색하여야만 한다. 둘째, 대화 인터페이스에 서로 상이한 내용의 대화 메시지가 수신될 때마다 대화 메시지를 프레젠테이션하기 위한 대화창의 잦은 이동은 대화자에게 대화 내용의 파악을 어렵게 한다. 여기서 대화창은 대화 인터페이스에서 대화 메시지를 프레젠테이션 하는 윈도우이다. 대화 메시지의 누적으로 전체 대화 메시지를 모두 나타낼 수 없

으므로 대화자가 대화 인터페이스의 슬라이더 바를 조정하여 대화창에 대화 메시지 프레젠테이션 할 부분을 변경하거나, 대화 메시지의 수신에 따라 수신된 대화 메시지를 나타내기 위하여 대화창에 프레젠테이션 되는 대화 메시지의 부분이 달라진다. 그리고 대화창은 대화 메시지가 누적될수록 대화창의 이동 빈도가 많아질 수 있다.

그러므로 본 논문에서는 위에서 언급한 트리 기반 대화 인터페이스의 문제점을 해결하여 응답 관계를 효율적으로 프레젠테이션 하는 트리 기반 대화 인터페이스를 제안한다. 제안한 트리 기반 대화 인터페이스는 텍스트 기반 대화 인터페이스의 장점과 XML 기반 메시지를 이용하여 응답 관계를 체계적으로 나타내면서, 최근 수신 메시지의 탐색, 응답 관계에 따른 대화창 이동의 최소화, 대화자 중심의 대화 인터페이스 관리를 지원한다.

본 논문의 구성은 다음과 같다. 제 2장에서는 대화 인터페이스에 관련한 연구에 대하여 기술한다. 제 3장에서는 제안한 대화 인터페이스의 설계에 대한 설명으로서 제안한 대화 인터페이스의 모델, 사용하는 메시지의 설계, 대화 인터페이스의 구동에 대하여 기술하며, 제 4장과 5장에서는 구현 및 결과와 앞으로의 연구 방향에 대하여 설명한다.

2. 관련 연구

대화에서 응답 관계를 지원하기 위한 텍스트 및 트리 기반 대화 인터페이스에 대한 연구로는 Vronay의 흐름 챗(Flow Chat)[7], Viegas의 대화 시각화(Visualizing Conversation)[2], M. Smith의 대화 트리(Conversation Tree)[4], Farnham의 리더 라인(Lead Line)[9] 등이 있다.

먼저, Vronay의 흐름 챗[7]은 대화 인터페이스에 각 사용자별로 트랙을 지정하고 대화자가 작성한 메시지를 시간별로 대화 인터페이스에 나타낸다. 각 대화자가 작성하는 대화 메시지는 대화 인터페이스에 표시된 자신의 트랙에만 나타나므로, 특정 시점에 누가 대화를 하고 있는지 나타내기 용이하다. 하지만 다자간 대화 메시지를 기술할 때 다중 응답 관계를 구분하여 나타내기 어렵다. Viegas의 대화 시각화[2]는 2D 기반의 대화 인터페이스를 사용하여 서클을 대화자로 표시하고, 대화 메시지를 수신할 때 서클의

크기를 대화 메시지의 길이에 맞춰 크기를 조정하여 나타내고, 시간의 경과에 따라 글자와 서클의 크기와 명도가 감소하도록 나타낸다. 그리고 대화 메시지의 기록을 위한 별도의 윈도우를 사용하여 대화자별로 타임 라인에 따라 기록된 대화 메시지를 나타내지만, 대화에서 응답 관계를 나타내기는 어렵다. Smith의 대화 트리[4]는 트리 기반의 대화 인터페이스로서 트리의 각 노드는 대화 메시지를 나타내기 위하여 사용하며 노드간의 깊이는 대화 메시지의 계층적 응답 관계를 나타낸다. 이 트리 기반 대화 인터페이스는 여러 대화자간 대화 메시지들의 응답 관계를 나타내지만, 새로 추가되는 응답 메시지가 기존 응답 메시지의 누적에 따라 대응 메시지와 상대적으로 떨어진 위치에 나타날 수 있다. 또한, 새로운 대화 메시지가 추가될 때마다 그 대화 메시지를 나타내기 위한 대화창의 이동으로 대화 내용을 파악하게 어렵게 만든다. Farnham의 리더 라인[9]은 텍스트 기반 대화 인터페이스에 대화를 이끄는 스크립트를 제공하는 대화 인터페이스이다. 리더 라인은 그룹의 의사 결정을 지원하기 위하여 대화 진행자 및 토론자 역할을 지정하고, 대화 진행자가 제시한 스크립트(script)에 따라 각 대화자들이 대화를 진행하는 인터페이스이다. 리더 라인에서 스크립트 메시지는 자연스러운 대화를 중단시킬 수 있으며, 반복된 스크립트의 사용은 대화자가 스크립트를 무시함으로 인하여 대화의 진행을 방해할 수 있다. 이 대화 인터페이스도 텍스트 기반의 대화 인터페이스로서 응답 관계의 지원이 미흡하다.

그 외, 실제 사용 중인 대화 시스템인 세이클럽(SayClub)[15], 프리챌(FreeChal)[16], 토마토넷(TomatoNet)[17], 가챗(GaChat)[18], 매직챗(MagicChat)[19]의 대화 인터페이스는 대부분 텍스트 기반 대화 인터페이스를 응용하여 사용하고 있으며, HTML 태그(tag) 및 스크립트 언어를 이용하여 대화 메시지에 부분적인 사회 정보를 지원한다[11,12]. 이러한 대화 메시지는 메시지의 서체, 색깔, 크기 등의 대화자가 결정할 수 있도록 지원하지만, 대화자의 응답 관계를 지원하기는 어렵다. 매직챗[19]은 2차원 바탕 화면에서 사용자 캐릭터 이미지와 연관된 아바타를 이용하여 대화를 지원하지만, 대화 내용을 추적하기 위하여 추가적인 텍스트 기반 인터페이스를 함께 사용하고 있다. 또한 대화자의 존재감은 어느 정도 나타낼 수 있지만, 대화에서 응답 관계는 지원하기 어렵다.

대부분의 관련 연구에서 응답 관계를 나타내기 위한 여러 가지 노력을 하고 있으나 응답 관계의 프레젠테이션에 대한 연구는 여전히 미흡한 편이다. 그러므로 본 논문에서는 대화의 응답 관계를 효율적으로 나타내기 위하여 텍스트 기반 대화 인터페이스의 장점과 XML 기반 메시지를 이용하는 트리 기반 대화 인터페이스를 제안한다.

3. 제안하는 트리 기반 대화 인터페이스

본 장에서는 제안한 대화 인터페이스의 설계와 대화 인터페이스에서 사용하는 XML 기반의 메시지를 기술한다. 또한 대화 인터페이스의 구동과 타 대화 인터페이스와의 기능을 비교하여 설명한다.

3.1 사용자 인터페이스 설계

기존 온라인 대화를 위한 대화 인터페이스의 일반적인 구성은 대화자 목록, 대화 메시지 기록 및 표시 부분, 메시지 입력 부분으로 이루어진다[7]. 하지만 본 논문에서 제안하는 대화 인터페이스는 기존 트리 기반 대화 인터페이스에 추가로 텍스트 기반 대화 인터페이스를 결합한 형태로 이루어진다. 즉 트리 기반 대화 인터페이스에 추가적으로 텍스트 기반 대화 인터페이스가 함께 있는 대화 인터페이스이다.

일반적인 텍스트 기반 대화 인터페이스는 대화 시스템에서 대화 인터페이스에 도착하는 대화 메시지를 순서대로 나타낸다. 그러므로 대화 메시지를 수신하는 대화 인터페이스에서는 현재 어느 대화자가 최근에 대화 메시지를 작성하여 발송했는지 확인하는 것이 용이하다. 그러나 트리 기반 인터페이스에서는 이러한 파악이 어렵다. 왜냐하면 대화에 참여한 여러 대화자가 대화 메시지를 동시에 발송할 때, 트리 기반 대화 인터페이스에서 수신되는 대화 메시지의 기록 위치는 응답 관계에 따라 여러 위치에서 산재되어 나타난다. 이러한 트리 기반 대화 인터페이스에서 대화 메시지들의 응답 관계는 체계적으로 나타내지만, 대화 메시지가 누적되면 트리 구조가 커지고 어떤 대화 메시지가 최근에 수신한 대화 메시지의 파악하기가 더욱 어렵다. 이러한 단점으로 대화자는 트리 기반 대화 인터페이스에서 트리의 어떤 부분에서 대화가 이루어지고 있는지 파악하기 위해서 전체 트리를 탐색해야하는 어려움이 있다. 그리고 이미 많은

응답 메시지가 많이 누적되어 있는 상황에서 새로운 대화 메시지가 대화 인터페이스에 수신될 때, 기존 트리 기반 대화 인터페이스는 응답 메시지와 그에 대응하는 대화 메시지와 근접 정도는 더욱 멀어짐으로써 진행되는 대화의 내용 파악이 어려워질 수 있다. 이에 따라 새로운 메시지가 수신될 때마다 그 대화 메시지를 나타내기 위한 대화창의 이동도 잦아 대화 내용의 파악이 더욱 어렵다.

기존 트리 기반 대화 인터페이스의 이러한 단점을 해결하기 위하여 제안하는 트리 기반 대화 인터페이스는 다음과 요구 사항을 만족하여야 한다. 첫째, 최근 수신한 대화 메시지의 탐색, 둘째, 응답 메시지와 대응 메시지와 상대적 근접성의 최소화, 셋째, 대화자가 관심을 가지는 대화 메시지 위주로 프레젠테이션 하는 대화 인터페이스 적응적 관리이다. 이러한 요구 사항을 만족시키기 위하여 제안한 대화 인터페이스의 구성은 네 부분으로 구성한다. 즉, 대화자 목록 인터페이스, 트리 기반 대화 인터페이스, 텍스트 기반 대화 인터페이스, 대화 종료 확인 인터페이스로 구성한다. 대화자 목록 인터페이스는 대화자가 대화에 참여와 탈퇴에 따라 자동적으로 갱신한 대화자 목록을 프레젠테이션 한다. 트리 기반 대화 인터페이스는 대화 메시지를 나타내는 단말 노드와 응답 관계를 나타내는 중간 노드로 구성한다. 텍스트 기반 대화 인터페이스는 대화 메시지 기록과 대화 메시지의 응답 관계에 관련된 대화 메시지 ID, 대화 메시지 입력창으로 구성한다. 대화 종료 확인 인터페이스는 대화 인터페이스의 종료를 위한 확인 버튼으로 구성한다.

다음 그림 1은 제안한 대화 인터페이스의 구성을 나타내는 문맥 표현(contextual representation)이다. 즉, 대화 인터페이스는 4개의 인터페이스로 구성되

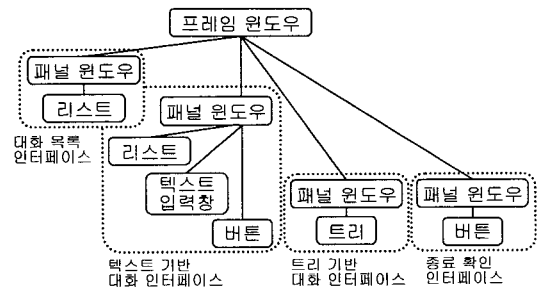


그림 1. 제안한 대화 인터페이스의 문맥 표현

며, 대화자 목록 인터페이스는 사용자 목록 및 접속 시간을 표시하는 리스트 객체를 포함하며, 텍스트 기반 대화 인터페이스는 대화 메시지가 서버에 도착하는 순서대로 기록하는 리스트 객체, 대화자의 입력을 받아들이는 텍스트 입력 객체, 입력 버튼 객체를 포함한다. 트리 기반 대화 인터페이스는 트리 구조를 나타내기 위한 트리 객체를 포함하며, 대화 종료 확인 인터페이스는 확인 버튼 객체를 포함한다.

3.2 메시지의 설계

제안한 대화 인터페이스에서 사용하는 메시지는 XML 기반의 메시지이다. XML 기반 메시지는 정보 표현을 위한 자연스런 형태를 제공한다[8,12]. 즉, XML 기반의 메시지는 메시지가 포함한 속성의 변경과 추가가 편리하다. 기존 온라인 대화에서는 대화 메시지를 단순히 스크립트 및 HTML 태그를 이용하여 대화 인터페이스에 사용하지만, 본 논문에서 제안한 대화 인터페이스에서는 XML 기반 메시지를 이용하여 더 많은 융통성과 정보를 지원한다.

제안한 대화 인터페이스에서 사용하는 메시지는 대화 메시지 및 제어 메시지로 구분되며, 대화 메시지는 대화자가 상대방 대화자와 대화를 위하여 사용되는 메시지이고, 관리 메시지는 대화를 지원하기 위하여 대화 인터페이스 및 대화 서버에서 사용하는 메시지이다. 먼저, 대화 메시지는 텍스트 및 트리 기반 대화 인터페이스에서 프레젠테이션 되는 각각의 대화메시지들은 대화자가 작성한 메시지와 함께 두 대화 인터페이스 상호간에 일대일 대응 관계를 가질 수 있는 정보를 포함한다. 대응 관계의 정보는 대화자가 트리 기반 대화 인터페이스에서 최근에 수신한 대화 메시지를 탐색할 때, 텍스트 기반 인터페이스의 최근 수신 대화 메시지를 이용하여 쉽게 파악하도록 사용의 편리성을 제공한다. 즉, 텍스트 기반 대화 인터페이스는 수신하는 모든 대화 메시지의 누적은 스택의 구조를 이용하기 때문에 항상 탑(top)에 위치한 대화 메시지가 최근에 수신한 대화 메시지가 된다. 이러한 특징을 이용하여 트리 기반 대화 인터페이스에서 구현하기 어려운 최근 수신 대화 메시지의 탐색을 텍스트 기반 대화 인터페이스를 이용하여 확인 후 그에 대응하는 트리 기반 대화 인터페이스를 자동으로 탐색하도록 한다. 즉, 트리 기반 대화 인터페이스에서 최근 수신 대화 메시지의 탐색을 위하여 대화

메시지는 이러한 두 대화 인터페이스에서의 대화 메시지 간 대응 관계 정보를 가져야한다. 대응 관계 정보에는 트리 기반 대화 인터페이스의 트리 구조에서 대응하는 대화 메시지가 위치한 노드로부터 루트 노드까지의 경로 정보를 포함하며, 이러한 경로 정보를 대화 메시지에 포함하기 위한 대화 메시지의 형식은 다음 DTD (Document Type Definition)와 같다.

```
<!DOCTYPE MESSAGE [
  <!ELEMENT MESSAGE (PID)>
  <!ELEMENT PID EMPTY>
  <!ATTLIST MESSAGE from CDATA #REQUIRED>
  <!ATTLIST MESSAGE to CDATA #REQUIRED>
  <!ATTLIST MESSAGE id CDATA #REQUIRED>
  <!ATTLIST MESSAGE reply CDATA #REQUIRED>
  <!ATTLIST PID pid0 CDATA #REQUIRED>
  .....
  <!ATTLIST PID pidn CDATA #REQUIRED>
]>
```

여기에서, 엘리먼트(element) 노드 MESSAGE는 대화자가 작성하는 대화 메시지를 위한 텍스트 노드와 대화 메시지의 응답 관계를 제공하는 엘리먼트 노드 PID로 구성되며, 엘리먼트 노드 MESSAGE는 속성으로 발송자(from), 수신자(to), 메시지번호(id), 응답 번호(reply)를 가지며, 엘리먼트 노드 PID는 응답 관계를 지원하는 정보로서 현재 메시지의 상위 응답 메시지인 엘리먼트 노드 MESSAGE 노드들의 속성인 ID 값(메시지 번호)들을 가진다. PID의 값 $pid_n(n=0, \dots, n)$ 에서 n 은 응답 계층의 깊이로서, 트리 기반 대화 인터페이스의 트리 구조에서 부모 계층에 위치한 MESSAGE 노드의 개수이다. 즉, pid_n 의 값은 트리 기반 대화 인터페이스에서 응답 관계의 따른 새로운 응답 노드의 위치를 결정하는 값으로 사용한다. pid_n 의 값을 사용하여 트리 기반 대화 인터페이스에서 트리 구조의 각 노드에 해당하는 대화 메시지는 텍스트 기반 대화 인터페이스의 각 대화 메시지와 일대일 대응 관계를 가진다. 다음은 대화 메시지의 예를 나타낸다.

```
<MESSAGE from = "상훈" to = "규민" id="52"
reply="46"> 다음 모임은 언제인가요?
</PID pid0 = "36" pid1 = "18"> </MESSAGE>
```

관리 메시지는 대화를 지원하기 위하여 대화 인터페이스와 대화 서버간에 사용되는 메시지이다. 이러한 메시지는 대화자 로그인 및 로그 아웃, 대화 메시지

ID 값 확인 등의 관리 목적으로 사용한다. 다음 표 1은 관리 메시지의 기능과 형식을 나타낸다.

관리 메시지 중 1, 2번 메시지는 대화 인터페이스에서 대화 서버로 전달되는 메시지로서, 대화자명 등록과 대화 메시지에 ID 값을 부여하기 위하여 사용하는 메시지이다. 그리고 3, 4, 5번 메시지는 대화 서버에서 대화 인터페이스로 전달되는 메시지로 대화자 목록의 전달, 대화자 명의 중복 통지, 대화자 목록의 갱신을 위하여 사용되는 메시지이다.

3.3 대화 인터페이스의 구동

제안한 트리 기반 대화 인터페이스는 텍스트 기반 대화 인터페이스와 함께 연동되어 작동한다. 이러한 연동을 위하여 텍스트 기반 인터페이스의 각 대화 메시지는 트리 기반 인터페이스의 트리 구조에서 각 단말 노드의 대화 메시지와 일대일 연관 관계정보를 이용한다. 다음 그림 2는 제안한 대화 인터페이스의 구동 및 동작을 나타내는 UML 기반의 상태 다이어그램이다.

여기서 먼저, “트리 기반 대화 인터페이스 구동” 상태에 포함되는 “노드 선택 리스너 구동” 상태는 대화자가 트리 기반 대화 인터페이스에서 응답할 노드의 선택을 대기하는 상태이며, “노드 선택 리스너 정지” 상태는 대화자가 응답할 노드를 클릭하면 새로운 노드가 선택한 노드의 자식 노드로 추가되고, 현재 선택한 노드를 기준으로 형제 및 자식 노드들이 폴딩(folding)되지 않도록 마킹한다. “현재 노드 재마킹” 상태는 대화자가 다시 선택한 대화 메시지를 기준으로 자식 노드들이 폴딩되지 않도록 한다. 이것은 대화 창에서 메시지 프레젠테이션 변화를 감소시키기 위하여 선택한 노드로 재 마킹하는 상태이다. 즉, 마킹된 노드를 포함하지 않는 형제 노드와 그 자식 노드들은 응답으로 수신한 대화 메시지가 추가되더라도 트리 구조의 노드를 확장하지 않는다. 이렇게 함으로써 마킹되지 않은 노드의 자식으로 추가되는 응답 대화 메시지는 대화창에 프레젠테이션 되지 않는다. 마킹된 노드의 형제 및 하위 노드에 대해서만 폴딩되지 않도록 하여 대화창의 이동을 최소화하여

표 1. 관리 메시지의 형식과 기능 설명

순번	관리 메시지의 형식	기능 설명	전송 방향
1	<!DOCTYPE user [<!ELEMENT user (CDATA)> >	대화자가 대화 인터페이스를 통하여 대화 서버에 대화명을 등록	대화 인터페이스에서 대화 서버로 전송
2	<!DOCTYPE requestID [<!ELEMENT requestID EMPTY> <!ATTLIST requestID id #REQUIRED> >	대화 인터페이스의 트리 기반 인터페이스에서 대화자가 메시지 작성한 후 대화 서버로 발송하면, 대화 서버는 그 메시지에 해당하는 ID를 결정하기 위한 ID 값을 반환	
3	<!DOCTYPE users [<!ELEMENT users (user+)> <!ELEMENT user (CDATA)> >	대화 서버가 새로 등록된 대화자의 대화 인터페이스에게 현재 대화중인 대화자의 목록을 전송	대화 서버에서 대화 인터페이스로 전송
4	<!DOCTYPE nameInUse [<!ELEMENT nameInUse EMPTY> >	대화 서버가 대화자명이 이미 등록되어 있는 경우 클라이언트의 대화 인터페이스에게 대화명 중복을 통지	
5	<!DOCTYPE update [<!ELEMENT update (user)> <!ELEMENT user (CDATA)> <!ATTLIST update type (login logout) #REQUIRED> >	대화 서버가 새로 등록된 대화자명을 활성화중인 대화 인터페이스에 전달	

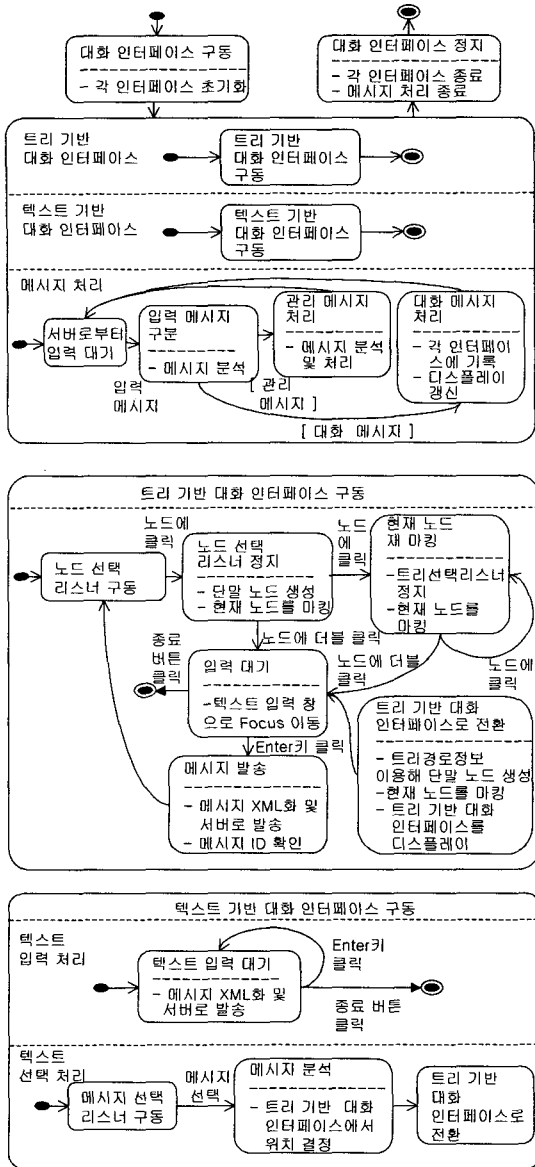


그림 2. 대화 인터페이스의 상태 다이어그램

대화자가 대화 내용의 파악을 용이하게 한다. “입력 대기” 상태는 생성된 노드에 포함된 텍스트 입력창에 직접 메시지를 작성할 수 있도록 포커스가 지정된 상태이며, “메시지 발송” 상태는 입력한 메시지를 대화 서버로 발송하며, 발송한 메시지의 ID를 대화 서버로부터 확인한다. “트리 기반 대화 인터페이스로 전환” 상태는 텍스트 기반 대화 인터페이스에서 대화자가 이미 수신된 대화 메시지를 선택할 경우, 그 메시지에 대응하는 트리 기반 대화 인터페이스 노드

의 응답으로 새로운 노드를 트리 기반 대화 인터페이스에 추가하고, 추가된 노드를 폴딩의 기준이 되도록 마킹한다. 이런 작업을 수행한 후 트리 기반 대화 인터페이스의 디스플레이를 갱신한다. “텍스트 기반 대화 인터페이스의 구동” 상태는 두개의 서버 상태 “텍스트 입력처리”와 “텍스트 선택 처리”를 가진다. “텍스트 입력 처리” 상태는 기존 텍스트 기반 대화 인터페이스에서 텍스트 입력과 입력 메시지를 XML화 하여 대화 서버로 발송하며, “텍스트 선택 처리” 상태는 대화창에서 대화 메시지 리스트에서 메시지를 선택할 수 있도록 하는 “메시지 선택 리스너 구동” 상태와 선택된 메시지에 포함된 트리 기반 대화 인터페이스에 대응하는 대화 메시지의 위치 정보를 확인한다. 확인된 정보를 가지고 “트리 기반 대화 인터페이스로 전환” 상태에 의하여 대화자의 인터페이스는 텍스트 기반 대화 인터페이스에서 트리 기반 대화 인터페이스로 이동한다.

트리 기반 대화 인터페이스의 문제점인 응답 메시지 간 근접 정도를 최소화하고, 대화자가 선택한 메시지를 중심으로 대화창의 이동을 최소화하기 위하여 응답 메시지를 최근 도착 순서대로 내림차순이 아닌 오름차순으로 정렬하여 대화자가 항상 최근에 수신된 응답을 쉽게 파악할 수 있도록 하여 항상 최근에 진행되는 대화의 내용을 파악하도록 한다. 이러한 오름차순 정렬은 대화창의 이동을 줄일 수 있으며, 트리 기반 대화 인터페이스에서 마킹에 의한 노드 확장의 감소와 함께 대화창의 이동을 최소화하여 대화자가 대화 내용을 파악하는데 용이하다. 다음 그림 3은 대화 메시지와 대화창의 일반적인 형태를 나타낸다. 그리고 다음 표 2는 그림 3과 같은 유사한 상황에서 기존 트리 기반 대화 인터페이스와 제안한 트리 기반 대화 인터페이스에서 대화창의 이동을 비교한 것으로 제안한 대화 인터페이스에서 대화창의 이동이 감소함을 나타낸다.

제안한 트리 기반 대화 인터페이스에서 사용하는

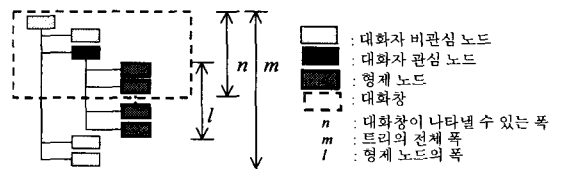


그림 3. 대화 창에서 대화 메시지의 형태

표 2. 대화창의 이동 비교 (단 $n < m$)

메시지의 형태 \ 인터페이스	기존 트리 기반 인터페이스	제안한 대화 인터페이스
대화메시지가 대화창안에 있고, $l \leq n$ 인 경우	$l = n$ 인 경우에 이동	이동 없음
대화메시지가 대화창에 걸쳐있는 경우	항상 이동	대화창 윗부분에 형제 노드가 걸쳐 있을 경우에만 이동
대화메시지가 대화창밖에 있는 경우	이동 있음	이동 있음

주요 프로시저들은 다음과 같다. 먼저 프로시저 `TextConversation()`는 텍스트 기반 대화 인터페이스에서 메시지 및 이벤트를 처리하는 프로시저이며, 프로시저 `TreeConversation()`는 트리 기반 대화 인터페이스에서 메시지 및 이벤트를 처리하는 프로시저이다. 그리고 프로시저 `ProcessingMessages()`는 트리 기반 인터페이스에서 대화 메시지 작성과 대화 서버로부터 메시지 수신을 처리하며, 프로시저 `ProcessingFolding()`은 트리 기반 대화 인터페이스에서 트리 구조의 노드 폴딩을 관리하는 프로시저이다. 프로시저 `TextConversation()`와 `TreeConversation()`는 대화 인터페이스에서 대화자의 대화 메시지 작성과 이벤트 처리를 지원하는 프로시저들이며, 프로시저 `ProcessingMessage()`는 트리 기반 대화 인터페이스가 대화 서버로부터 수신되는 대화 메시지를 처리하는 프로시저이다.

Procedure `TextConversation()`

```

{
  if (User inputted a message in text field) {
    Initialize attributes of the message;
    Attach the message to the text-based interface;
    Attach the message as children of root node on the tree-based interface;
  }
  if (User clicked a message on interface) {
    Decide a location for inserting a node on the tree-based interface as reply of a selected message using pid attribute's values of the selected message;
    Generate a node for reply on the tree-based interface;
    Transfer to tree-based interface and focus the added node;
    Wait user's input on the tree-based interface;
  }
}

```

Procedure `TreeConversation()`

```

{
  if (User clicked a node) {
    Attach a new_node below the clicked node;
    Wait an input of an user;
    The the new_node is set as a mark_node;
    // Default value of the mark_node is a root node of the tree structure.
  }
}

```

```

if (User finished writing a message) {
  Set up attributes of the message;
  Attach the message beneath last message on the text-based interface;
}
}
}
procedure ProcessingMessages()
{
  if (an message is inputted by the server) {
    if (the message is a conversational message) {
      Search an parent node using information of the message;
      Add a new_node below the searched the parent node;
      if (the tree-based interface is running) ProcessingFolding(new_node);
    }
  }
}
procedure ProcessingFolding(the_added_node)
{
  if ( the_added_node is a children or sibling of the marked node )
    Display content of the new node with the message;
}
}

```

다음 표 3은 기존 대화 인터페이스와 제안한 대화 인터페이스간 특성을 비교한 것이다. 제안한 대화 인터페이스는 기존 트리 기반 대화 인터페이스의 단점을 해결하기 위하여 트리 기반 인터페이스에 텍스트 기반 대화 인터페이스의 장점을 결합한 대화 인터페이스로서 응답 관계뿐만 아니라 트리 기반 대화 인터페이스에서 지원하기 어려운 최근 수신 메시지의 파악, 수신 메시지에 의한 산발적인 대화창의 이동과 응답 메시지 간 근접성을 최소화할 지원함으로써 대화 내용의 파악을 용이하게 한다.

4. 구현

제안한 트리 기반 대화 인터페이스는 JDK 1.4와 Kawa 5.0을 이용하여 구현하였다. 대화를 지원하는 대화 서버는 클라이언트/서버 구조로서 대화자 및 클라이언트의 관리를 위하여 DOM(Document Object Management) 트리를 사용하였으며, 대화 메시

표 3. 대화 인터페이스의 특성 비교

특성 \ 대화 인터페이스	흐름 챗[7]	대화 트리[4]	챗 씨클[2]	리더 라인[9]	제안한 대화인터페이스
응답 관계 체계성	△	○	×	△	○
최근 수신메시지 파악 용이성	○	×	○	○	○
관련 대화메시지 연결성	×	○	×	×	○
최근 응답 메시지간 근접성	△	△	×	△	○
응답 관계 기록 및 복원성	×	×	×	×	○

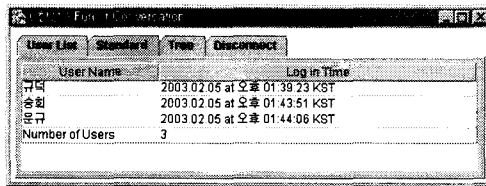
(○: 지원, △: 약간지원, ×: 미지원)

지와 관리 메시지를 구분하여 메시지의 전달 방향과 관리 메시지에 대한 응답과 클라이언트의 생성 및 소멸에 따른 자원 관리를 지원한다[13].

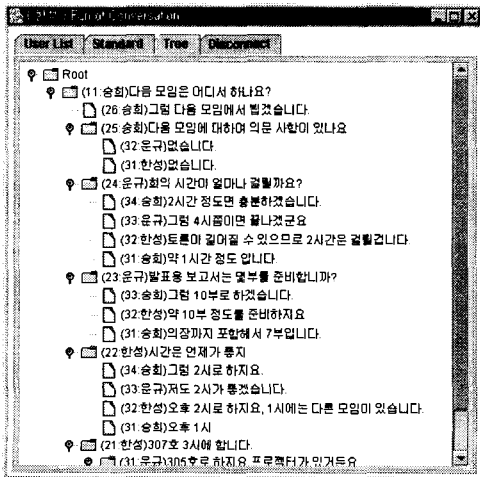
그림 4는 구현한 대화 인터페이스로서, 대화자 목록, 트리 기반 인터페이스, 텍스트 기반 인터페이스의 사용자 인터페이스를 나타낸다.

그림 4 (b)의 트리 기반 대화 인터페이스에서 대화 메시지의 앞의 괄호 안에 좌측 첫 번째 숫자는 대화 메시지의 응답 계층을 나타내며, 응답이 도착한 순서는 좌측 두 번째 숫자부터 나타나는 일련번호이다.

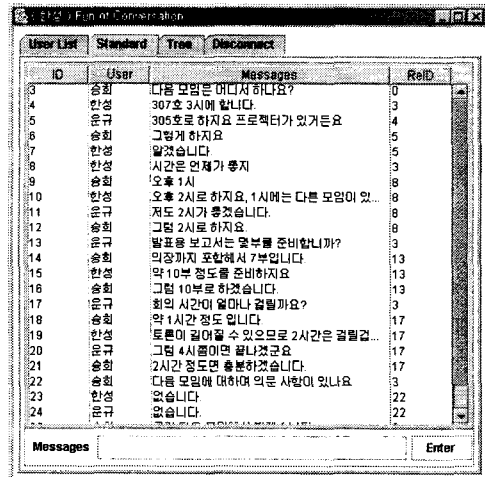
그리고 응답 계층의 깊이가 깊을수록 첫 번째 숫자 값이 크며, 대화 메시지와 응답 메시지의 응답 계층의 크기 차이는 1이 된다. 두 번째 숫자이후의 일련번호는 크면 클수록 최근에 수신한 메시지임을 나타낸다. 또한, 일련번호의 값은 현재 누적된 응답 메시지의 개수를 나타내는 것으로 얼마나 많은 응답 메시지가 있는지 파악이 용이하다. 이러한 오름차순에 의한 대화 메시지의 정렬로 추가되는 응답 메시지는 그에 대응하는 대화 메시지와 항상 가장 근접한 위치에 있으므로 대화창의 이동을 감소시킨다.



(a) 대화자 목록



(b) 트리 기반 대화 인터페이스



(c) 텍스트 기반 대화 인터페이스

그림 4. 제안한 트리 기반 대화 인터페이스의 사용자 인터페이스

5. 결 론

본 논문에서는 온라인 대화에서 응답 관계를 효율적으로 프레젠테이션 하는 트리 기반의 대화 인터페이스 설계와 구현에 대하여 기술하였다. 제안한 트리 기반 대화 인터페이스는 기존 대화 인터페이스에서 지원하기 어려운 최근 대화 메시지의 탐색, 대화 메시지의 응답 관계의 파악, 응답 메시지 간 근접성의 최소화, 대화 인터페이스 적응적 관리를 지원함으로써 사용의 용이성을 나타낸다. 또한 대화 및 관리 메시지를 XML 기반으로 작성함으로써 차 후 새로운 정보의 보완 및 관리를 융통성 있게 지원할 수 있으며, 대화 서버는 DOM을 사용하여 대화 시스템의 관리 및 유지 보수를 효율적으로 지원할 수 있다.

앞으로의 연구 방향은 대화자의 사회 정보에 대한 시각화이다.

참 고 문 헌

- [1] C. Charlton, C. Little, R. Lloyd, S. Morris, and I. Neison, "Good Business Practice Needs Good Communications - New Generation Chat Software for Real-time Discussion," Proc. of the 10th Int. Workshop on Database & Expert systems Applications, 1999.
- [2] J. Donath, K. Karahalios, and F. Viegas, "Visualizing Conversation," Proc. of the 32nd Hawaii Int. Conf. on System Sciences, pp.1-9, 1999.
- [3] S. Ortiz Jr., "Instant Messaging: No Longer Just Chat," IEEE Computer, Vol.34, No.3, pp.12-15, 2001.
- [4] M. Smith, J. Cadiz, and B. Burkhalter, "Conversation Trees and Threaded Chats," Proc. of the ACM2000 Conf. on CSCW, pp.97-105, 2000.
- [5] M. O. Thirunarayann and A. Perez-Prado, "Structured Chat," ACM Ubiquity, Vol.1, Issue 38, 2000.
- [6] M. O. Thirunarayann, "Cutting Down on Chat Confusion," ACM Ubiquity, Vol.1, Issue 45, 2001.
- [7] D. Vronay, M. Smith, and S. Drucker, "Alternative Interfaces for Chat," Proc. of the 12th Annual Symposium on User Interface Software and Technology, pp.19-26, 1999.
- [8] P. Vuorimaa and T. Heiszter, "XML Messaging," Proc. of the Senventh Int. Conf. on Distributed Multimedia System, pp.394-400, 2001.
- [9] S. Farnham, H. R. Chesley, D. E. McGhee, and R. Kawal, "Structured Online Interactions: Improving the Decision-Making of Small Discussion Groups," Proc. of the ACM 2000 Conf. on Computer supported Cooperative Work, pp.299-308, 2000.
- [10] J. Preece, "Supporting Community and Building Social Capital," Communication of the ACM, Vol.45, No.4, pp.37-39, 2002.
- [11] 김경덕, 류권열, "실시간 대화 행위에서 XML을 이용한 메시지의 시각적 표현", 한국정보과학회 HCI2002 학술발표논문집(CD).
- [12] 김경덕, "XML 기반의 대화 메시지 설계", 한국정보과학회 2002 봄 학술발표논문집(B), 제29권, 제1호, pp.574-576, 2002.
- [13] 김경덕, "대화에서 응답 관계의 시각화", 한국정보과학회 2002 가을 학술발표논문집(II), 제29권, 제2호, pp.226-228, 2002.
- [14] 김경덕, "트리 기반 대화 인터페이스에서 최근 수신 메시지의 프레젠테이션", 한국정보과학회 영남지부 제10회 학술발표논문집, 제10권, 제1호, pp.85-89, 2002.
- [15] SayClub, <http://www.sayclub.com>
- [16] FreeChal, <http://www.freechal.com>
- [17] TomatoNet, <http://www.tomatonet.com/chat2>
- [18] GaChat, <http://www.gachat.com>
- [19] MagicChat, <http://www.mhouse.net/mcworld>



김 경 덕

1989년 경북대학교 자연대(이학사)

1991년 경북대학교 컴퓨터공학과 (공학석사)

1999년 경북대학교 컴퓨터공학과 (공학박사)

1991년~1996년 (주)웨스트시스
템 연구소 연구원

2000년~현재 위덕대학교 컴퓨터멀티미디어공학부 조
교수

관심 분야 : 멀티미디어 프레젠테이션, 컴퓨터 언어, 원격
교육 등