

트리기반 대화 인터페이스에서 메시지의 효율적인 프레젠테이션

김경덕

위덕대학교 컴퓨터멀티미디어공학부
e-mail: kdkim@uu.ac.kr

Efficient Presentation of Messages on Tree-based Conversation Interface

Kyungdeok Kim

Division of Computer and Multimedia Engineering
Uiduk University

요 약

본 논문에서는 트리기반 대화 인터페이스에서 대화 메시지의 효율적인 프레젠테이션 방법을 제안한다. 대화 인터페이스는 교육, 오락, 비즈니스 등에서 유용한 도구로서 활용이 되고 있지만, 형식적인 대화를 지원하기 위한 방법이 아직 미흡한 편이다. 기존 트리기반 대화 인터페이스는 응답 관계를 체계적으로 나타낼 수는 있지만 메시지가 수신될 때마다 메시지의 창의 이동이 발생하며, 또한 다양한 메시지의 주제에 따라 다중으로 수신되는 메시지의 프레젠테이션으로 인한 메시지 류의 이동으로 대화 내용의 파악이 어렵다. 그러므로 본 논문에서는 트리기반 대화 인터페이스에서 대화 메시지에 해당하는 노드의 폴딩(folding)을 이용하여 효율적인 프레젠테이션을 위한 방법을 제안한다. 응용 분야로는 게임, 협업, 원격교육 등이다.

1. 서론

온라인 대화는 교육 및 게임 등에서 많이 활용되고 있으며, 이동 통신과 연동하여 다양한 서비스 기술들이 개발되고 있다. 현재 많은 대화 인터페이스는 이차원 및 삼차원의 그래픽 표현을 사용하여 사용자에게 사용의 편리성을 제공하고 있으나 기존 대화 메시지 및 인터페이스의 형식을 벗어나지 못하고 있다[1, 2, 3].

일반적으로 이용되는 대화 인터페이스는 텍스트기반, 트리기반, 아바타(avatar)기반의 인터페이스가 있다. 텍스트기반 대화 인터페이스는 주로 텍스트 메시지를 이용하며 사용자가 자신의 다양한 의사를 기술하기는 쉽지만, 대화 메시지를 수신하는 대화 인터페이스에 나타나는 대화 메시지들은 대화 시스템에 대화 메시지가 전달되는 순으로 프레젠테이션되어 상대방 사용자는 대화 내용의 파악이 어렵다. 트리기반 대화 인터페이스는 앞에서 언급한 텍스트기반 대화 인터페이스는 대화 메시지들의 응답 관계를 계층적인 트리 구조로 나타냄으로서, 텍스트기반

대화 인터페이스의 단점인 응답 관계를 명확히 나타낼 수 있다. 하지만 메시지가 수신될 때마다 메시지 창의 이동이 발생할 수 있으며, 수신되는 여러 메시지가 응답 관계에 따라 프레젠테이션 되어야 할 위치가 여러 곳에 산재됨으로서 현재 진행 중인 대화 내용을 파악하기 어렵다[4, 5, 6]. 아바타기반 대화 인터페이스는 사용자의 캐릭터를 이용하여 대화 인터페이스에서 사용자를 구분하여, 사용자가 자신의 존재감을 다른 대화 인터페이스에 비하여 상대적으로 나타내기 용이하다. 하지만 대부분은 대화 내용 파악을 위하여 텍스트기반 대화 인터페이스를 부분적으로 이용한다. 형식적인 대화 내용을 프레젠테이션하기 위한 대화 인터페이스는 일반적으로 트리 구조를 채택하여 활용하기 있지만 위에서 기술한 트리기반 대화 인터페이스의 단점으로 인하여 사용이 미흡한 편이다.

그러므로 본 논문에서는 기존 트리기반 대화 인터페이스에서 대화 메시지의 수신에 따른 효율적인 프레젠테이션을 위한 방법을 제안한다. 본 논문의

구성은 다음과 같다. 제 2절에서는 관련 연구에 대하여, 제 3절에서는 트리기반 대화인터페이스에서 메시지 폴딩 정책에 대하여, 제 4절에서는 폴딩에 의한 대화 메시지의 프레젠테이션에 대하여, 제 5절에서는 결론을 기술한다.

2. 관련 연구

대화 인터페이스를 개선하기 위한 관련 연구로는 Conversation Tree[4], Chat Spaces[7], Flow Chat [8], Lead Line[9] 등이 있다.

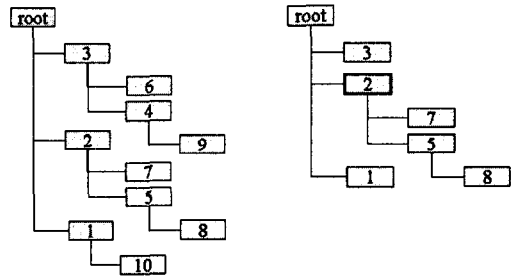
먼저, Conversation Tree[4]는 트리기반의 대화 인터페이스에서 대화의 응답 관계를 나타낸다. 이 대화 인터페이스는 사용자간 메시지들의 응답 관계를 명확히 나타내지만, 최근 수신한 대화 메시지를 프레젠테이션하기 위한 메시지 뷰의 잦은 이동으로 대화 내용의 파악이 어렵다. Chat Spaces[7]는 대화의 응답 관계를 나타내는 대화 맵(map)을 이용하여 대화 내용을 나타내며, 대화 메시지는 순차적 흐름의 메시지 뷰와 응답 관계를 나타내는 트리 뷰를 가진다. 순차적 흐름의 메시지 뷰에서 사용자의 선택에 의하여 트리 뷰에 관련된 메시지의 응답들을 나타낸다. 이러한 메시지 프레젠테이션은 사용자가 선택한 메시지에 관련된 질의응답만을 나타냄으로서, 관련한 다른 사용자의 응답에 대한 메시지는 파악하기 어렵다. Flow Chat[8]은 각 사용자별 타임 트랙을 지정하고 각 사용자가 작성한 메시지를 시간별로 인터페이스에 나타낸다. 각 사용자는 자신의 트랙에만 메시지를 작성하므로, 대화 메시지 간에 전체적인 응답 관계를 구분하여 나타내기 어렵다. Lead Line[9]은 텍스트기반 인터페이스에 대화를 이끌어가는 스크립트를 지원하는 인터페이스이다. 이 인터페이스는 그룹의 의사 결정을 지원하기 위하여 대화 진행자 및 토론자 역할을 지정하고, 지정된 역할과 대화 진행자가 사전에 제시한 스크립트에 따라 대화를 이끌어가는 인터페이스이다. 하지만 지원되는 스크립트 메시지가 사용자간 자연스러운 대화를 중단시킬 수 있으며, 스크립트의 반복된 사용은 사용자가 스크립트를 무시할 수 있어 대화의 진행이 어려워질 수 있다.

대화 메시지의 체계적인 프레젠테이션을 위한 여러 연구가 이루어지고 있으며, 전체 대화 내용의 체계적인 표현을 지원하기 위한 대화 인터페이스에 대한 연구는 아직 미흡한 편이다. 그러므로 본 논문에서는 트리기반 대화 인터페이스에서 트리 구조의 각 노드에 해당하는 메시지를 사용자 선택에 따라 비관 메시지는 폴딩 되도록 하여 대화 메시지를 효율적으로

나타낸다.

3. 트리기반 대화 인터페이스에서 폴딩 정책

트리기반 대화 인터페이스는 대화 메시지들 간의 응답 관계를 계층적인 트리 구조를 이용하여 대화 내용을 나타낸다. 트리기반 대화 인터페이스는 트리 구조의 특성에 따라 각 단말 노드 부분이 수신되는 대화 메시지의 위치이다. 즉 각 단말 노드의 하위 레벨의 위치가 응답으로 수신될 대화 메시지의 위치이다. 이러한 이유로 트리기반 대화 인터페이스에서는 메시지를 수신할 때마다 메시지 뷰의 이동의 파악과 현재 진행 중인 대화 내용의 파악을 이해하기 어렵다. 그러므로 본 논문에서는 대화 인터페이스에서 사용자가 선택하는 관심 메시지를 중심으로 그 하위 노드와의 다른 노드들은 폴딩 되도록 인터페이스를 설계한다. 다음 그림 1은 트리기반 대화 인터페이스에서 대화 메시지의 일반적인 프레젠테이션과 폴딩에 의한 프레젠테이션을 나타낸다.



(a) 일반적인 프레젠테이션 (b) 폴딩에 의한 프레젠테이션

그림 1. 대화 메시지의 프레젠테이션

그림 1에서 각 숫자는 대화 메시지가 작성된 일련번호를 나타낸다. 여기서 일반적인 프레젠테이션은 메시지의 기존 트리기반 대화 인터페이스의 메시지 프레젠테이션을 나타낸다. 즉, 3번 메시지의 응답은 6, 4번 메시지이며, 4번 메시지의 응답은 9번임을 나타낸다. 폴딩에 의한 프레젠테이션은 사용자가 선택한 주제를 가진 노드의 자식 노드들만 보이고 다른 노드들은 폴딩 되도록 한다. 이러한 폴딩 정책은 대화 인터페이스의 대화 메시지 수신에 의한 메시지 뷰의 이동을 감소시키고 대화 내용의 파악을 용이하게 한다. 즉, 2번 메시지의 응답에 대한 메시지들은 프레젠테이션 되지만, 3번과 1번 메시지의 응답 메시지의 프레젠테이션은 나타나지 않는다. 이러한 폴딩 정책은 화면 크기가 작은 대부분의 대화 인터페이스

이스에서 메시지 뷰의 이동을 감소시키며, 사용자의 대화 내용 파악을 용이하게 한다. 또한 각 계층별 메시지를 내림차순으로 정렬하여 수신되는 메시지를 각 계층별 최상위 위치에 놓아줌으로서 메시지 뷰의 이동을 기존 트리기반 대화 인터페이스와 비교할 때 상대적으로 이동을 감소시킨다.

4. 대화 메시지의 프레젠테이션

트리기반 대화 인터페이스에서 대화 메시지의 효율적인 프레젠테이션을 위한 폴딩 정책을 지원하는 메시지의 구성은 다음과 같다. 즉, <메시지, 발신자, 수신자, 일련번호, 부모 노드의 메시지 일련번호들> 6개의 요소를 가진 튜플이다. 여기서, 부모 노드의 메시지 일련번호들은 트리 구조에서 대화 메시지의 기록 위치를 결정하기 위하여 사용한다.

다음 프로시저들은 트리기반 대화 인터페이스에서 사용자의 메시지와 이벤트 처리, 및 폴딩 처리 과정을 기술한 것이다.

```
// procedure for the Tree-based interface
Procedure TreeConversations()
{
  if (User clicked a node) {
    Attach a new_node below the clicked node;
    Wait for an input of a speaker;
    Set the new_node as a mark_node;
    // The mark_node has a root node of the tree
    // as a default value
  }
  if (User finished writing a message)
    Set up attributes of the message;
}

// procedure for messages received from server
procedure ProcessingMessages()
{
  if (an message is received from server) {
    if (the message is a conversational message) {
      Search for a parent node using information
        of the message;
      Add a new_node below searched parent node;
      if (the tree-based interface is displaying)
        ProcessingFolding(new_node);
    }
  }
}

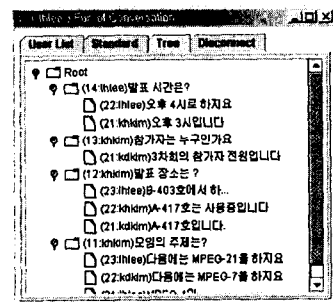
// procedure for folding strategy
procedure ProcessingFolding(the_added_node)
{
  if ( the_added_node is a children or sibling of the
    mark_node )
    Display content of the_added_node with the message;
  } else {
    Fold the_added_node;
    Do not present the_added_node in order to reduce
      a change of message view;
  }
}
}
```

기존 트리기반 대화 인터페이스와 폴딩 정책을 가진 트리 기반 대화 인터페이스의 특징을 비교하면 다음과 같다. 먼저, 메시지 뷰 안에 모든 메시지가 포함되어 있더라도 기존 대화 인터페이스는 관심 메시지의 자식 메시지의 크기(개수)와 메시지 뷰의 크기가 같으면, 메시지 뷰의 이동이 발생한다. 하지만 폴딩 정책을 가진 대화 인터페이스는 내림차순 정렬에 의하여 메시지 뷰의 이동이 발생하지 않는다. 또한 사용자 관심 메시지의 자식 메시지들이 메시지 뷰에 걸쳐 있을 경우 기존 대화 인터페이스는 이동이 발생하며, 폴딩 정책을 가진 대화 인터페이스는 단지 메시지 뷰의 상단에 메시지가 걸쳐 있을 경우에만 메시지 뷰의 이동이 발생한다.

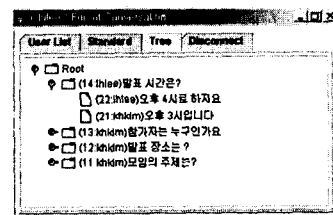
이와 같이 기존 대화 인터페이스의 특성에 비해 폴딩 정책을 가진 대화 인터페이스가 상대적으로 메시지 뷰의 이동을 감소시킬 수 있다. 이러한 특성으로 인하여 대화 인터페이스에서 메시지를 기존 대화 인터페이스에서보다 효율적으로 프레젠테이션 한다.

5. 결론

폴딩 정책을 적용한 트리기반 대화 인터페이스의 구현은 Java 1.4와 Kawa 5.0를 이용하였으며, 다음 그림 2는 구현한 대화 인터페이스를 나타낸다.



(a) 일반적인 대화 인터페이스



(b) 폴딩정책을 가진 대화 인터페이스

그림 2. 대화 인터페이스

그림 2의 대화 인터페이스에서 (a)는 일반적인 트리 기반 대화 인터페이스에서 볼 수 있는 인터페이스를 나타내며, (b)의 인터페이스는 폴딩 정책으로 인한 메시지 뷰의 내용을 축약된 형태로 나타낸다. 각 메시지 앞부분에 기술된 괄호 속의 내용은 숫자 부분과 사용자 이름으로 구성되어 있으며, 숫자 부분에서 앞부분은 트리의 계층 구조에서 레벨을 나타내고 뒷부분은 수신되는 메시지의 일련번호를 나타낸다. 이러한 폴딩 정책은 메시지 뷰의 이동을 감소시켜 대화 내용을 효율적으로 프레젠테이션 함을 보였다.

앞으로의 연구 방향은 유비쿼터스 환경에서 적합한 대화 인터페이스의 개발이다.

참고문헌

- [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] M. O. Thirunarayann, "Cutting Down on Chat Confusion," ACM Ubiquity, Vol. 1, Issue 45, 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] K. Kim, "Minimization of Jumpy Quality of Display by Locations of Replied Turns on Tree-based Conversational Interface," Proc. of the 30th KISS Spring Conf., Vol. 30, No. 1, pp. 585-587, 2003.
- [6] K. Kim, "Conversation Interface supporting Relations among Turns," Proc. of 2003 HCI, Vol. 12, No. 1, pp. 617-621, 2003.
- [7] W. Geyer, A. J. Witt, E. Wilcox, and M. Muller, "Chat Spaces," Proc. of DIS2004, pp. 333-336, 2004.
- [8] 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.
- [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.