

## 협업 방식의 문제 해결을 위한 그룹원 간의 구조적 대화 지원 시스템

전건호\* · 서용무\*

### Structured Group Interaction System for Collaborative Problem Solving

Geonho Jeon\* · Yongmoo Suh\*

#### ■ Abstract ■

These days we come across many wicked problems whose solutions are beyond individuals intellectual ability. These problems can be resolved through collaborative group interaction. We developed an Internet-based asynchronous group interaction support system, after looking into the collaborative problem solving process and the IBIS (Issue-Based Information System) argumentation model. It has the following characteristics : 1) it is developed based on the modified IBIS model which is a model for group interaction to resolve wicked problems; 2) it supports both processes of seeking and comparing solutions, while most similar systems do not have a feature to support the latter process; 3) different structures can be defined dynamically according to the purpose of group interaction, so that it could be used for collaborative problem solving in a specific domain. To show the usability of the system, we carried out an experiment, whose result is shown at the end.

*Keyword : Structured Group Interaction, Collaborative Problem Solving, Wicked Problem*

## 1. 서론

하나의 공통된 목표를 달성하기 위해 많은 사람들이 함께 공동 작업을 하는 경우가 적지 않다. 예를 들면, 그룹 회의나 그룹 프로젝트, 또는 팀에 의한 소프트웨어 개발 등에서의 공동작업이 그 예이다. 이러한 공동작업 참여자의 목적은 의사결정이나 복잡한 개념의 정립, 설계, 컴퓨터 알고리즘, 부서의 예산 할당 등 각각의 상황에 따라서 달라지게 된다. 그러나 공동작업의 공통점은 목적을 달성하기 위해 상호 대화가 필요하다는 것이며, 이 상호 대화를 협업이라고 한다[14].

문제 해결을 위해 협업을 이용하는 이유는 복잡한 문제를 해결하기 위해 요구되는 경험이나 지식 등을 각 개인 한 사람이 충분히 가지고 있지 않기 때문이다. 각 개인은 어떤 한 전문분야에서 전문가가 될 수 있지만 모든 분야에서 전문가가 될 수는 없기 때문이다. 협업 접근 방식의 장점은 협업의 참여자가 같은 문제를 다른 관점에서 검토하고 분석하여 문제를 완전히 이해하고, 결국 최적의 솔루션을 찾아 낼 수 있다는 데 있다.

협업에서 참여자는 다른 참여자의 식견이나 아이디어, 질문, 그리고 비판 등을 필요로 한다. 그들은 아이디어 생성이나 교환, 지식의 추출이나 검증 같은 다양한 협업 활동으로부터 나온 결과를 공유하거나, 이해를 위한 공유 공간 생성 등을 필요로 하게 된다. 그래서, 협업은 협업 환경 내에서 작업할 때나 협업을 돕는 컴퓨터나 정보기술을 사용하는 곳에서 가능하게 된다. 즉, 효과적인 협업을 위해서 협업 어플리케이션을 가진 기반 기술이 필요하다. 이메일이나 게시판, 채팅 서비스 등의 인터넷 기본 서비스가 협업을 지원하기는 하지만, 협업 어플리케이션으로 보지는 않는다. 왜냐하면, 이 서비스들은 느린 피드백 문제와 대화 내용의 섞임 문제[8]뿐만 아니라 공통 관심사의 문제에 대한 공유 기반을 제공하지 않기 때문이다. 협업을 위해서 우리는 참여자의 대화를 지원하고 공유 이해 공간을 지원하는 협업 어플리케이션을 필요로 하게 된다.

본 논문은 협업 방식의 문제 해결을 지원하는 비동기식 다자간 대화 어플리케이션 시스템(ShallWeTalk)의 개발과 실험 결과를 보여준다. 협업과 협업 시스템(다음 장에서 자세히 소개함) 관련 이론과 모델 및 기존 연구들을 바탕으로 프로토타입 시스템을 개발하였다. 본 연구는 다자간 대화가 어떤 정해진 룰에 의해 이루어 져야 된다는 결과를 보여주며, 그렇지 않으면 다자간 대화의 목적을 달성하는데 큰 효과가 없다는 것을 보여준다. 그래서, 다자간 대화를 이끌 의사소통행위(communicative act) 요소들을 이용하여 다자간 대화를 위한 대화 룰을 정의하였다.

ShallWeTalk 시스템의 특징은 다음과 같다. 첫째, 대화 룰에 의해 이루어지는 다자간 대화가 트리 구조로 보여지게 된다. 그러므로, 참여자가 장시간이 지난 후에도 이미 이루어진 논쟁간의 구조를 쉽게 파악 할 수 있다. 둘째, 다자간 대화가 솔루션 찾기(솔루션 제안을 위한)와 대부분의 기존 시스템에서 지원되지 않는 솔루션 평가(제안된 솔루션의 평가) 순서로 진행된다. 셋째, ShallWeTalk 시스템은 사용자가 그들의 협업 목적에 따라서 동적으로 대화 룰을 정의 할 수 있도록 지원함으로써 다양한 협업 환경에서 사용될 수 있도록 설계되었다.

본 논문의 구성은 다음과 같다. 제 2장에서 기존 연구에 대한 결과를 몇몇 그룹으로 분류하여 보여준다. 제 3장에서는 ShallWeTalk 시스템에 대해 소개하고, 제 4장에서 ShallWeTalk 시스템과 일반 게시판 시스템을 비교하기 위한 유용성 실험과 실험 결과를 기술한다. 그리고, 제 5장에서 본 논문의 결과를 정리하고 향후 연구방향을 제시한다.

## 2. 다자간 대화 시스템

협업은 효과적인 의사소통이나 다자간 대화를 통해서 가능하며, 기본적으로 다자간 대화를 지원하는 많은 협업 어플리케이션의 개발 시도가 있었다. 현재, 이메일과 채팅 시스템이 의사소통의 주 수단 이긴 하지만 최근에 개발되는 대부분의 시스템은

웹기반으로 만들어지며 다양한 스타일의 다자간 대화를 지원한다. 아래에서 보인 시스템들은 협업을 위해 프로젝트 관리나 정책 만들기 또는 설계 같은 많은 다른 영역에서 접근을 시도하며, 협업 방식의 문제 해결을 위한 수단으로 사용될 수 있는 시스템들이다.

다자간 대화 지원 시스템은 구조화되는 정도에 따라 몇몇 그룹으로 분류될 수 있다. 첫 번째 그룹으로 일반적인 채팅이나 THREADED CHAT[15] 시스템이 있다. 그러나, 이 시스템들은 협업 방식의 문제 해결에는 적당하지 않다. 왜냐하면, 이 시스템들을 이용하는 대화는 어떤 한 주제에 집중하거나 대화 흐름 자체를 이해하기에 쉽지 않으며, 특히 많은 아이디어가 대화에서 혼재할 때 어려움은 더욱 커진다.

두 번째 그룹으로 스크립트 기반의 시스템이 있다[7, 10]. 예를 들면, LEAD Line[7] 시스템은 채팅 같은 온라인 대화에서 구조의 부재가 몇몇 문제의 근원임을 인식하고 대화의 질을 증진시키기 위해 대화에 상황 종속 스크립트를 추가하였다. 시스템의 스크립트 지원 기능이 집중화 동기 대화를 지원하고 따라서 대화를 통한 컨센서스 성취를 돕는다고 하였다. 그러나, 스크립트 기반의 시스템은 너무 구체적인 상황을 위해 스크립트를 정의하기 때문에 다양한 일반적인 목적의 협업 문제 해결을 위해 사용하기에는 적당하지 않다. Pfister의 연구[10]에서 두 학습 프로토콜(Learning Protocol)은 설명을 위한 논쟁과 협동 방식의 텍스트 요약 논쟁을 위한

스크립트로 정의 되었다.

세 번째 그룹은 의사소통행위의 세트를 지원하는 시스템이다[1, 2, 19]. C-CHENE 시스템은 태스크 기반과 반응 대화를 위해 유연한 구조화 기능을 가지고 있다[1]. 이 시스템은 의사소통행위의 제한된 세트를 지원하지만 이 세트의 사용에 대한 강제성은 없다. 또한, 이 시스템은 어떤 구체적인 상황에서 의사소통행위 순서에 대한 제약과 가이드를 제공한다. 의사소통행위들은 대화 내에서의 그 기능에 따라 몇몇 카테고리로 분류된다. 그러므로, 훨씬 이해하기 쉽도록 구조화된다. Coordinator는 분산 의사소통 시스템에서 대화를 제어할 수 있도록 설계된 시스템이다[19]. 이 시스템의 사용자는 의사소통행위의 제한된 세트가 주어지며 그 순서는 증가된 전환 네트워크 문법(예, request/promise, offer/accept, 등등)들에 의해서 정의된다. 그러나, 단지 의사소통행위의 한 세트를 제공하는 것만으로 공유된 이해를 생성하는 문제의 솔루션이 될 수는 없으며 그것들이 구조화되어야 한다.

네 번째 그룹은 Stahl의 연구[16-18]에서 보인 것처럼 프로세스를 위한 구조를 제공한다. WebGuide는 웹 기반의 쓰레드 논의 시스템으로 협업 방식의 지식 구축 모델을 기반으로 개발되었다[17]. 이 시스템은 개인, 그룹, 및 비교 관점을 가진 지식 구축 프로세스를 위한 구조를 제공한다. 그리고 이 시스템은 연결 구조를 이용하여 쓰레드 방식 논의 시스템의 공통적인 확산(divergence) 문제를 해결하려고 시도하였다. 또한, 협상 컴포넌트를 사용함으로써

〈표 1〉 다자간 대화 시스템과 ShallWeTalk 시스템의 특징

시 스템	구 조 화	연구적용분야	동기성
Threaded Chat	쓰레드 채팅	의사결정	동 기
LEAD Line	스크립트(role)	의사결정	동 기
Learning Protocol	스크립트(rule)	학 습	동 기
C-CHENE	의사소통행위	학 습	동 기
Coordinator	의사소통행위	커뮤니케이션	동 기
WebGuide	프로세스 구조	학 습	비동기
ShallWeTalk	의사소통행위, 프로세스 구조	문제해결, 의사결정	비동기

써 개인 관점이 그룹 관점이 될 수 있다. 비록, 이 시스템이 위에 보인 것과 같은 많은 장점을 가지고 있지만 사용자들이 관점들 간의 관계를 이해하는 것은 쉽지 않다. 그리고 만약 논의의 양이 한 스크린에 보이기에 힘들 정도로 커진다면 연결 구조를 공유된 이해 구축을 위한 장애물인 확산 문제를 위한 솔루션으로 사용하기에 적당하지 않다.

다자간 대화 시스템과 ShallWeTalk 시스템의 특징을 <표 1>에 표시하였다.

### 3. ShallWeTalk : 구조적 비동기 다자간 대화 시스템

앞 장에서 검토한 것처럼 지금까지 다양한 다자간 대화 지원 시스템이 개발 되었다. 이 시스템들은 다자간 대화 내용 분석[4-6]과 관련된 이론이나 모델[6, 16]을 기반으로 하여 설계 되었다. ShallWeTalk 시스템은 과학적 커뮤니티 내에서 협업 방식의 문제 해결 프로세스를 연구하고 토론을 위해 IBIS(Issue-Based Information System) 모델을 기반으로 설계 되었다.

#### 3.1 과학적 커뮤니티 내에서 협업 방식의 문제 해결 프로세스

협업 이론의 기본 주장은 사람들이 개개인의 능력을 넘어서는 복잡한 문제 해결을 위해 함께 작업해야 한다는 것과 사람들이 더 효과적이고 효율적으로 함께 작업 할 수 있도록 협업 어플리케이션이 지원될 필요가 있다는 것이다[14].

Stahl의 연구[17]에서 보인 협업 방식의 지식 구축의 모델은 개인적 이해와 사회적 지식 구축의 두 사이클로 이루어진다. 학습은 개인의 이해로부터 시작되어 다른 사람들과의 대화라는 사회적 프로세스를 통해 문화를 형성하고, 이는 다시 개인 이해를 위한 하나의 자원으로 제공되게 된다. 즉, 개인에 의해 생성된 개인적 믿음이 의사소통이나 논의, 설명 및 협상 같은 사회적 대화를 통해서 사회적/문화적 지식이 되는 것이다. 그러한 방식으로

얻어진 지식은 나중에 질문과 재해석을 받게 된다. 또한, 이 논문은 컴퓨터 지원이 요구되는 모델 내에서의 주요 요점(문제 해결을 통한 지식 습득과 쌓인 지식의 요약 정보 제공 등)들을 제안하였다. 개인들과 그룹간의 상호작용은 협업의 주장과 일치한다[16].

앞에서 보인 협업 방식의 지식 구축 모델은 과학적 지식 획득 프로세스와 유사하다. 과학자들은 계속해서 그들의 협업 방식을 통한 연구 결과를 지식으로 계속해서 발표하게 된다. 여기에서 협업은 한 문제에 대한 솔루션을 얻기 위해 함께 작업하는 과학자들로 이루어진 한 그룹이나 같은 문제에 관심이 있는 과학자들로 이루어진 다른 그룹들간에서 모두 일어난다는 것을 알 수 있다. 후자에서 연구 결과의 충돌이 일어나는 경우 올바른 한 결과를 선택하거나 또는 새로운 하나의 연구 결과로 종합하기 위해 협업을 필요로 하게 된다. 그러므로, 과학은 변증법적인 프로세스를 통해서 진보한다고 말할 수 있다[11]. 즉, 한 개인의 연구에 의한 하나의 새로운 이론(anti-thesis)은 기존에 받아들여진 많은 이론(theses)들과 비교되고 필요할 경우 다시 새로운 이론(synthesis)으로 변경된다. 새로운 이론은 거절되거나 부분적으로 수정되거나, 또는 종합 프로세스를 거쳐 완전히 새로운 이론(thesis)으로 형성 될 수 있다. 여기에서 새 이론과 기존의 이론들 간의 상호작용<sup>1)</sup>이 과학적 진보를 위해서 필수적이며 같은 연구분야에 있는 모든 연구팀들은 그 분야 내에서 과학적 진보를 위해 협업을 하게 된다. 협업 방식의 지식 구축 프로세스와 위에서 언급한 과학적 지식 획득 프로세스를 모두 협업 방식의 문제 해결 프로세스로 볼 수 있다. 이 프로세스는 주어진 문제에 대해 사람들이 개인이 찾을 수 있는 것보다 더 나은 솔루션을 찾을 수 있도록 함께 작업을 하는 것이며 이는 다시 두 개의 부 프로세스(솔루션 찾기와 솔루션 평가)로 이루어진다. 솔루션 찾기 부 프로세스에서는 한 문제가 주어지

1) 과학적 진보를 위한 과학자들 간의 협업의 한 형태로 보일 수 있다.

고 그 대안 솔루션들이 제안되며 지지나 반대 의견들의 지원을 받게 된다. 솔루션 평가 부 프로세스에서는 각 대안 솔루션은 주어진 의견들의 타당성이 체크됨으로써 평가가 이루어지게 된다. 한 솔루션을 지지 또는 반대하는 의견은 질문과 답을 제공하게 되거나 정보의 요청과 요청에 대한 응답 등에 의해서 타당성을 평가받게 된다. 대화 중에 이 두개의 부 프로세스간에 명확한 구별은 없다. 즉, 몇몇 사람들이 솔루션 찾기 부 프로세스에 참여하는 동안 다른 몇몇 사람들은 평가 부 프로세스에 참여할 수 있다.

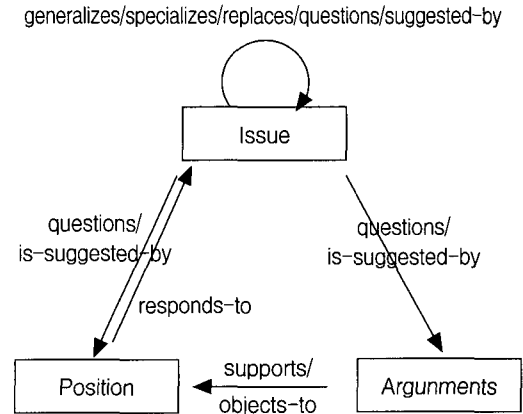
### 3.2 IBIS(Issue-Based Information Systems) 모델

IBIS는 논쟁이론 기반 위에서 고안된 모델이다 [13]. 난해한 문제가 성공적으로 해결 되도록 하기 위해 의견이 이 모델의 구조에 의해 제어 되도록 한다. 난해한 문제란 절대적으로 옳거나 그른 답이 있는 것이 아니라 좀 더 좋거나 또는 나쁜 솔루션을 가진 복잡한 문제이다. <그림 1>은 IBIS 모델의 구조를 보여준다. 이 구조에 따르면 한 이슈(Issue)에 대한 대화는 구조에 의한 쌍(pair)의 의사소통행위 요소로 이루어진 룰에 따르는 많은 가지를 가진 트리 형태로 보일 수 있다. 예를 들면, 한 이슈는 많은 포지션(Position)을 가질 수 있고, 각 포지션은 많은 의견(Argument)을 가질 수 있다. 여기서, 이슈와 포지션은 의사소통행위의 한 쌍이고, 포지션과 의견도 마찬가지로 한 쌍이다.

IBIS 모델의 장점은 그 구조가 무척 간단해서<sup>2)</sup> 어떠한 도메인의 어떠한 이슈에서도 쉽게 적용할 수 있다는 것이다. 이 모델의 또 다른 장점은 아주 명백한 의사소통을 지원한다는 것이다. 이러한 장점들 때문에 IBIS 모델은 대부분의 도메인에서 협업 방식의 문제 해결을 위해 적합한 솔루션이 될

2) IBIS 모델의 구조가 만약 모든 가능한 링크를 사용하면 복잡할 수 있는데 실제로는 단지 그 구조의 부분(예, 이슈-포지션-의견)만 사용된다.

수 있다. IBIS를 기반으로 하는 시스템들이 기존 연구[3, 9, 12]에서 소개되고, 또 개발되었다.



<그림 1> IBIS 모델의 구조

그러나, IBIS 모델도 부족한 점이 있다. 예를 들면, IBIS 모델은 솔루션 찾기 프로세스는 잘 지원하지만 솔루션 비교 프로세스는 잘 지원하지 못한다. 주어진 문제(이슈)에서 IBIS 모델의 사용자가 문제에 대한 솔루션을 제안(솔루션 찾기)하게 되고 각 솔루션에 지지나 반대 의견을 추가(솔루션 비교)하게 된다. 그렇지만, 각 의견의 타당성에 대한 검증은 지원되지 않는다. 즉, 협업 방식의 문제 해결을 위해 중요한 솔루션 평가가 지원되지 않는 것이다.

### 3.3 다자간 대화를 위한 문제의 특성

협업에서 다루게 되는 문제의 특성은 <표 2>에서 보인 것처럼 크게 1) 정형화(formalization)와 2) 동기화(synchronization)로 분류해 볼 수 있다. 정형화 영역에는 다시 1) 잘 정형화되고 폐쇄적인 좀 더 절차적인 정형화된 도메인과 2) 명확히 정의된 목적이 없고, 문제에 대한 명확한 해답도 없으며, 또한 정확한 솔루션을 구성하는 것에 대한 판단을 하기 위한 명확한 기준이 없는 비 정형화 된 도메인으로 구분해 볼 수 있다. 앞에서 언급한 난해한 문제는 비 정형화 된 도메인의 문제라고 할 수 있

다. 이 비 정형화 된 도메인을 Cook[5]은 오픈 엔디드 도메인(open-ended domain)으로 정의하였다. 둘째, 동기화 영역에는 1) 동시에 참여자의 협업 작업이 필요한 동기 도메인과 2) 참여자의 동시 협업 작업이 필요 없는 비 동기 도메인으로 나누어 볼 수 있다.

<표 2> 협업을 위한 문제 도메인

정형화 동기화	정형화된 도메인	비 정형화된 도메인
동기 도메인	(1)	(3)
비 동기 도메인	(2)	(4)

<표 2>에 따르면 솔루션을 찾는 어느 한 문제는 (1) 동기-정형, (2) 비 동기-정형, (3) 동기-비정형, 그리고 (4) 비 동기-비 정형의 네 클래스 중의 하나에 포함이 되게 된다. 각 클래스의 문제 특성은 다음과 같다.

- (1) 클래스의 문제 : 답이 명확하며 한 그룹이 짧은 시간동안의 동기식 대화를 통해서 해결 할 수 있는 문제이며 대화의 내용이 장기간 꼭 저장 될 필요는 없다.
- (2) 클래스의 문제 : 답이 명확하며 한 그룹이 긴 시간동안의 비동기식 대화를 통해서 해결 할 수 있는 문제이며 대화의 내용이 장기간 저장 될 필요가 있다.
- (3) 클래스의 문제 : 답이 명확하게 정의되지 않으며 한 그룹의 짧은 시간동안의 동기식 대화를 통해서 해결 하고자 하는 문제이며 대화의 내용이 장기간 꼭 저장 될 필요는 없다.
- (4) 클래스의 문제 : 답이 명확하게 정의되지 않으며 한 그룹이 긴 시간동안의 비동기식 대화를 통해서 해결 하고자 하는 문제로 대화의 내용이 장기간 저장 될 필요가 있다.

(1)과 (3)의 클래스 문제에서는 동기식 대화 중에 답을 찾을 수 있도록 하기 위해 그룹 구성원의 대화가 원활하게 유지되도록 하는 기능의 지원에

포커스를 맞추어야 하지만, (2)와 (4) 클래스 문제에서는 비동기식 대화가 일어나기 때문에 대화의 내용이 저장되어야 하고 또 쉽게 기존 대화의 내용을 이해하거나 분석할 수 있도록 하는 기능의 지원에 포커스를 맞추어야 한다. 특히, (4) 클래스 문제는 장기간의 비동기식 다자간 대화를 통해서 문제를 해결하게 된다.

본 연구에서 제시한 ShallWeTalk 시스템은 특히 (4) 클래스의 비 정형화된 난해한 문제를 장기간에 걸쳐 비 동기 방식으로 그룹원 들간의 대화를 통해서 해결할 수 있는 기능을 지원하게 된다.

### 3.4 ShallWeTalk 시스템

협업 방식의 문제 해결 프로세스와 IBIS 모델의 연구를 통해서 구조적 다자간 대화 지원 시스템인 ShallWeTalk 시스템을 설계하고 개발하였다. 이 시스템은 다음의 설계 요구 사항을 지원한다. 1) 단순화 된 구조를 지원한다. 2) 공유 공간에서 대화의 부분 검색을 지원한다. 3) 구조의 동적인 정의를 지원한다. 4) 응답에 대한 알림 기능을 지원한다. 5) 다자간 대화 내용의 현 상태에 대한 요약 내용을 보여준다. 아래에서 위 요구사항들에 대해 좀 더 자세히 설명한다.

#### 3.4.1 솔루션 찾기와 평가 프로세스 모듈을 위한 단순화된 구조

IBIS 모델의 분석을 통하여 이 모델의 복잡성이 비록 논쟁 트리(tree)가 세 종류의 대화 요소(예, 이슈, 포지션, 의견)로 이루어지지만 그들을 연결하는 많은 다른 종류의 링크에서 나온다는 것을 알았다. 예를 들면, IBIS 모델에 따라 이슈는 다른 이슈를 생성하거나 구체화하거나 교체하거나 질문을 하거나 또는 제안한다는 것이다. 즉, 두 이슈 사이의 관계는 굉장히 다양 할 수 있으며 이를 한 모니터의 제한 된 공간에서 보이기에 한계가 있다. 결국, 긴 시간 동안의 논쟁 트리는 논의되고 있는 문제에 대한 이해를 어렵게 만드는 긴 가지를 가진 덩굴(bushy) 형태가 될 수 있다.

위 문제를 해결하기 위해서 각 요소 쌍들간의 단 순화된 링크를 가진 논의 요소를 정의하였다. 각 쌍 은 한 요소가 다른 한 요소를 가지는 형태의 한 롤 이 된다. 본 논문에서 두 개의 롤 세트를 정의하였 는데 하나는 솔루션 찾기 프로세스를 위한 것이고 다른 하나는 솔루션 평가 프로세스를 위한 것이다.

첫째, 아래에 P(문제), SP(부 문제), S(솔루션), SA(지지 의견) 및 OA(반대 의견) 등의 대화 요소 들을 이용하여 솔루션 찾기 프로세스를 위한 대화 롤을 정의하였다. “이슈-포지션-의견” 구조에 따르 면 P(문제)와 SP(부 문제)는 이슈가 되고 S(솔루 션)은 포지션이며 SA(지지 의견)과 OA(반대 의견) 은 의견이 된다.

- $P \rightarrow SP$  : 한 문제가 복잡할 경우 여러 개의 부 문제들로 나누어진다.
- $P|SP \rightarrow S$  : 문제나 부 문제는 여러 개의 대안 솔루션을 가진다.
- $S \rightarrow SA|OA$  : 한 솔루션은 여러 개의 지지나 반대 의견들을 가진다.

둘째, 아래에 At(공격), Rq(요청), Q(질문) 등의 대화 요소들을 이용하여 솔루션 평가 프로세스를 위한 대화 롤을 정의하였다.

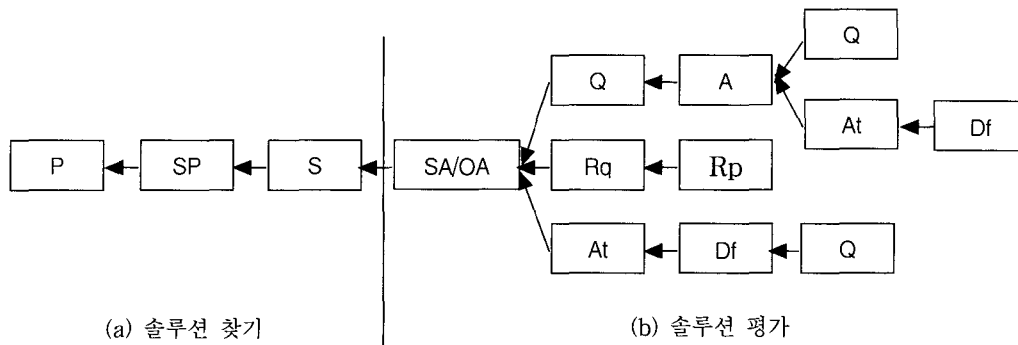
- $SA|OA \rightarrow At|Rq|Q$  : 각 의견은 공격이나 증 거나 예 같은 정보의 요청을 가지거나 또는 좀 더 자세한 설명에 대한 질문을 가질 수 있다.
- $At \rightarrow Df$  : 공격은 타당성을 제공하기 위한 방

어를 가질 수 있다.

- $Rq \rightarrow Rp$  : 요청은 적당한 증거나 예를 가질 수 있다.
- $Q \rightarrow A$  : 질문은 답변을 가질 수 있다.
- $Df|Rp|A \rightarrow At|Rq|Q$  : 각 방어, 요청, 응답 은 각각 공격, 응답, 그리고 필요할 경우 질문을 가질 수 있다.

위 롤은 다자간 대화에서 <그림 2>와 같은 논 쟁 트리를 생성 할 수 있다. 솔루션 찾기 프로세스 에서 사용자는 주어진 문제를 필요할 경우 몇 개의 부 문제로 나눌 수 있다. 그들은 지지나 반대 의견 들을 가진 솔루션들을 가진다. 솔루션 평가 프로세 스에서 사용자들은 각 의견의 타당성을 체크함으 로써 나아가 각 대안 솔루션을 평가하게 된다. 여 기서 타당성 체크는 질문을 하거나 다른 정보를 요 청하거나 의견에 대한 공격 등으로 이루어진다. 같 은 종류의 가치가 여러 번 올 수가 있다.

위에서, IBIS 모델에서 생성된 의사결정 트리와 ShallWeTalk 시스템에서 생성된 트리와의 차이점 이 있음을 알 수 있다. ShallWeTalk 시스템의 의 사 결정 트리는 다자간 대화 내용을 쉽게 이해 할 수 있게 지원하는 트리와 같은 레벨에서 솔루션 찾 기 프로세스를 위한 논의 요소들을 가진다. 반면, IBIS 모델의 트리는 한 가지에 같은 종류(예, 이슈) 의 긴 가지나 논의 요소를 가질 수 있으므로 협업 방식의 문제 해결을 위해 사용자가 다자간 대화에 참여하는 것을 방해 할 수 있다.



<그림 2> ShallWeTalk 시스템의 구조

3.4.2 공유 공간에서 대화의 부분 검색

ShallWeTalk 시스템에서 논쟁 트리는 Oracle DBMS 내의 데이터베이스에 저장된다. 그리고, 이 데이터베이스는 다자간 대화의 어느 참여자(ShallWeTalk 시스템에 등록된 사용자)도 접근할 수 있다. 예를 들면, 그들은 대부분 지지 의견을 가진 솔루션이나 반대 의견보다 지지 의견을 많이 가진 솔루션을 검색 할 수 있다. ShallWeTalk 시스템은 내부적으로 SQL 명령어 생성 인터페이스를 가지고 있고 사용자가 이용하게 된다. SQL 명령은 Oracle 시스템으로 전달되고 그 결과가 ShallWeTalk 시스템으로 다시 전달되며 최종적으로 사용자를 위해 모니터에 표시되게 된다. 그리고, DBMS의 동시(concurrency) 제어 기능을 사용함으로써 공유 공간을 한 사람 이상이 동시에 접근함으로써 생길 수 있는 문제를 방지 할 수 있다. 현재, 사용자 인터페이스가 대부분의 SQL 명령을 지원 할 수 있을 정도로 지원되지는 않는다. 앞으로 좀 더 개선 될 필요가 있다.

3.4.3 동적인 구조의 정의

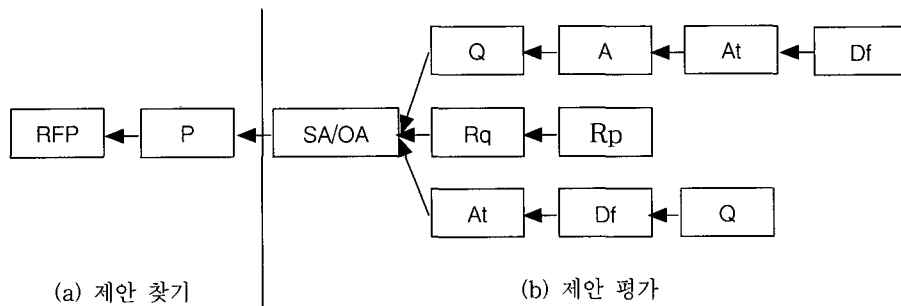
ShallWeTalk 시스템의 또 다른 중요한 장점은 협업 방식의 문제 해결을 위한 다양한 상황에 맞도록 동적으로 다양한 구조들을 정의 할 수 있다는 것이다. 앞에서 설명한 단순화된 구조는 일반적인 목적의 협업 방식의 문제 해결을 위한 다자간 대화에서 사용될 수 있다. 그러나, 특정한 영역의 다자간 대화에 사용하기에는 적당하지 않다. 다자간 대

화의 어떤 특정한 목적에 따라서 다른 구조(즉, 의견 요소들과 그들 간의 링크)가 다자간 대화를 시작하기 전에 정의 될 필요가 있으며, ShallWeTalk 시스템은 IBIS의 기본 구조인 “이슈-포지션-의견”에 맞춰 다양한 구조를 시스템 내에서 동적으로 만들고 그 구조에 따라서 그룹원들 간의 대화가 이루어질 수 있게 한다.

소프트웨어 공급자를 선택하는 도메인을 예로 들어 보면 다음과 같다. 한 회사가 그 회사를 위한 시스템 개발을 할 소프트웨어 공급자를 선택 할 때 다음의 단계를 거치게 된다. 1) 많은 공급자에게 RFP(Request For Proposal → Proposal)를 배포한다. 2) 그들로부터 제안을 받게 된다. 3) 몇몇 제안자들로부터 제안에 대한 설명을 듣게 된다. 4) 제안들을 평가하고 비교한다. 5) 최종적으로 RFP의 요구 사항에 가장 적합한 한 공급자를 선택하게 된다.

앞의 3)과 4)단계에서 다자간 대화를 위한 한 구조를 정의할 필요가 있다면 다음과 같이 정의 할 수 있다. “이슈-포지션-의견” 구조에 따르면 RFP(문제)는 이슈가 되고 P(솔루션)은 포지션이며 SA(지지 의견)과 OA(반대 의견)는 의견이 된다.

- RFP → P(Request For Proposal → Proposal) : 한 RFP 품이 많은 제안서를 가질 수 있다.
- P → SA | OA(Proposal → Supporting-Argument or Objecting-Argument) : 한 제안서가 많은 지지나 반대 의견을 가질 수 있다.
- SA | OA → At | Rq | Q
- At → Df



<그림 3> 소프트웨어 공급자 선정을 위한 구조



- $Rq \rightarrow Rp$
- $Q \rightarrow A$
- $Df | Rp | A \rightarrow At | Rq | Q$

위 구조에서 마지막 다섯 개 쌍의 의미는 앞 절에서 설명한 것과 같다. <그림 3>은 소프트웨어 공급자의 선정을 위한 구조의 예를 보여준다. ShallWeTalk 시스템은 동적으로 기존의 구조를 수정하거나 새로운 구조를 등록함으로써 쉽게 새로운 구조를 정의할 수 있게 지원한다.

#### 3.4.4 응답에 대한 알림 기능

협업 방식으로 난해한 문제를 해결하기 위한 다자간 대화는 일반적으로 장시간이 걸린다(특히, ShallWeTalk 시스템과 같이 비동기 방식인 경우). 그러므로, 다른 사람이 응답 글을 올린 경우 알림 기능을 제공함으로써 다시 응답을 하거나 또는 하지 않는 결정을 빠르게 할 수 있도록 지원 할 수 있다. ShallWeTalk 시스템은 다른 사람이 응답 글을 올렸을 때 빠른 피드백을 위해 자동적으로 이메일을 보내게 된다.

#### 3.4.5 다자간 대화 내용의 현 상태에 대한 요약

구조적 대화 트리에 많은 노드가 있을 때 새로운 사용자는 다자간 대화의 현 상태를 파악하기가 쉽지 않다. 예를 들면, 제안된 많은 대안 솔루션 중에서 어느 것이 가장 많이 지지 받는 솔루션인지를 파악하기가 쉽지 않다. ShallWeTalk 시스템은 각 대안 솔루션에 대한 plausibility number와 각 의견에 대한 validity number를 보여줌으로써 다자간 대화의 현 상태를 쉽게 파악 할 수 있게 지원한다. validity number는 “이슈-포지션-의견” 구조의 포지션을 지원하는 각 의견에 대한 평가 단계의 그룹 대화를 바탕으로 수치화 한 값으로 각 의견의 타당성 값을 보여준다고 할 수 있다. plausibility number는 포지션에 대한 의견들의 타당성 값을 취합한 값으로 주어진 이슈에 대해 각 포지션이 어느 정도나 적합한지를 보여준다. 즉, plausibility number가 클수록 주어진 문제에 대해 적합한 솔루션

이라고 할 수 있다. 각 값은 다음과 같이 계산된다.

- Validity number : 각 의견에 대한 Validity number의 계산은 다음과 같이 한다. 1) 만약, 의견이 가치를 하나도 가지지 않으면 0값을 가진다. 2) 의견이 가지들을 가지면 각 가지의 validity number값을 합한 값이 된다. 2-1) 한 가지의 끝 노드(leaf)가 Df, Rp 또는 A 노드이면 모든 공격 의견이 방어가 된 것이기 때문에 validity number 값은 1이 된다. 2-2) 반대로, 한 가지의 끝 노드가 At, Rq 또는 Q 노드이면 -1의 값을 가지게 된다.
- Plausibility number : 한 솔루션에 대한 plausibility number는 음의 값이 아닌 validity number를 가진 지지 의견의 수에서 음의 값이 아닌 validity number를 가진 반대 의견의 수를 뺀 값이 된다(음의 validity 값을 가진 의견은 타당성이 없기 때문에 고려를 하지 않는다).

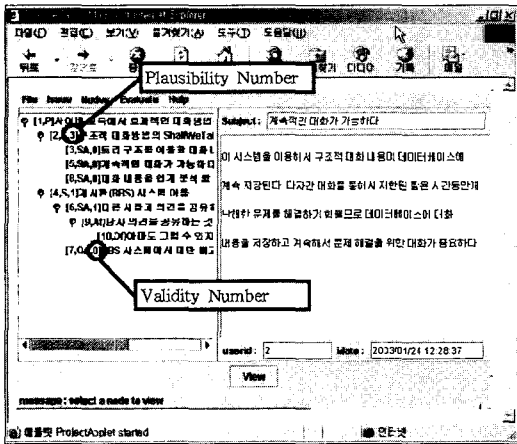
따라서, 대화 트리가 많은 노드를 가진 덩불처럼 되더라도 사용자는 앞의 두 값들로부터 다자간 대화가 어떻게 진행되고 있는지를 쉽게 파악할 수 있다. 그 값들은 새로운 노드가 트리에 생성될 때마다 자동적으로 갱신된다.

### 3.5 ShallWeTalk 시스템을 사용한 구조적 다자간 대화의 한 예

시스템 개발은 자바 컴파일러(J2SDK v1.4.1)를 이용하여 애플릿(applet) 프로그램으로 개발하였기 때문에 미리 다운로드 하여 설치 할 필요가 없으며, 자바 플러그인(JRE-1.4.0 이상, Java Runtime Environment)이 설치된 어느 웹 브라우저에서도 이용이 가능하다. 다자간 대화의 내용은 데이터베이스 서버(Oracle 9i release 1(9.0.1.1.1) for Windows)에 저장 되도록 개발하였다.

<그림 4>는 ShallWeTalk 시스템을 이용하여 다자간 대화가 어떻게 진행되는지를 보여준다. 그림에서 트리구조 부분의 표현 방식은 입력한 일련

번호와 대화 요소(P : 이슈, S : 포지션, SA : 지원 의견, OA : 반대 의견, At : 공격, 등등) 및 주제 형식으로 표현하였다. 포지션일 경우 plausibility number, 의견일 경우 validity number를 함께 표시하였다. 주어진 문제는 “사이버 교육에서 효과적인 대화 방법은 무엇인가?”(P1 ; 그림에서 트리구조 부분의 일련번호 1번)이고 두 개의 대안 솔루션(S1과 S2 ; 트리구조 부분의 일련번호 2번과 4번)이 주어졌다. <그림 4>에서 보인 것처럼, 첫 번째 대안 솔루션은 세 개의 지지 의견(SA1, SA2, SA3 ; 트리구조 부분의 일련번호 3번, 5번, 그리고 8번)을 가지고 두 번째 대안 솔루션은 한 개의 지지 의견(SA4 ; 트리구조 부분의 일련번호 6번)과 한 개의 반대 의견(OA1 ; 트리구조 부분의 일련번호 7번)을 가졌다. 현재의 시점에서 BBS 시스템보다 Shall-WeTalk 시스템이 더 나은 것으로 논의되고 있다. BBS 시스템이 대안 솔루션들을 비교하는 것이 어렵기 때문이다. 타당한 논의 진행처럼 보이는 이 결과는 다음 장의 결과 분석에서 증명된다.



<그림 4> ShallWeTalk 시스템을 이용한 대화의 한 예

### 4. 유용성 테스트

ShallWeTalk 시스템이 사용자에게 유용한지를 테스트하기 위해 ShallWeTalk 시스템과 BBS 시스템을 비교하는 유용성 테스트를 실시하였다.

### 4.1 테스트 방법

**참여자 :** MIS를 전공하는 K 대학교 대학원의 석사과정 16명을 선택하였다. BBS 시스템을 많이 사용해 봤으며 ShallWeTalk 시스템은 테스트를 하면서 처음으로 사용하였다.

**실험 방법 :** 먼저, 참여자에게 1시간 30분 동안 ShallWeTalk 시스템의 사용법에 대해 설명하고 16명의 참여자를 두 개(A, B)의 그룹으로 나누었다. 그리고 BBS 시스템과 ShallWeTalk 시스템을 사용하면서 협업 방식으로 주어진 두 개의 질문을 해결하게 하였고 대면 대화는 금지되었다. 주어진 두 개의 문제는 다음과 같다. 1) 대선에서 어느 후보가 가장 좋은 대통령 후보인가? 2) 서울의 교통 혼잡 문제를 어떻게 해결 할 수 있을 것인가? 각 참여자는 다음과 같은 과업을 수행하였다.

- 1) 실험 1 : 두 그룹에게 첫 번째 질문이 주어졌으며, 그룹 A와 그룹 B는 각각 BBS와 Shall-WeTalk 시스템 하나씩을 사용하게 하였고 기간은 1주일이었다.
- 2) 실험 2 : 두 그룹에게 두 번째 질문이 주어졌으며 사용 시스템을 교체하였다(즉, 그룹 A가 실험 1에서 BBS 시스템을 사용하였다면 ShallWeTalk 시스템을 사용함). 역시, 기간은 1주일이 주어졌다.
- 3) 설문조사 : 모든 참여자가 두 시스템의 사용성과 효과성을 비교 할 수 있는 두 세트의 설문에 참여하였다.
- 4) 개선을 위한 질문 : 모든 참여자에게 “Shall-WeTalk 시스템의 개선점은 무엇인가?”라는 오픈 질문이 주어졌다.

### 4.2 테스트 결과

이 절에서는 실험 1과 실험 2의 다자간 대화 내용 분석과 설문 내용 분석 결과를 보인다.

#### 4.2.1 다자간 대화 내용 분석

<표 3>는 실험에서 다자간 대화 내용의 분석

<표 3> 다자간 대화 내용의 분석 결과

	실험 1		실험 2	
	BBS	ShallWeTalk	BBS	ShallWeTalk
전체 의견 수	24	21	24	14
대안 수	7*	3	7	4
주제의 의견 수	8	0	2	1

주) \* 3개의 중복 내용 의견 포함.

결과를 보여준다. 두 실험에서 BBS 시스템의 사용자가 ShallWeTalk 시스템의 사용자보다 더 많은 의견과 대안을 올렸다. 아마 BBS 시스템에 더 익숙한 때문인 것으로 해석된다. 그러나 주목할만한 사항은 BBS 시스템에서 주제에 벗어나는 의견이 더 많이 올라 왔으며 7개 의견 중 3개는 내용이 거의 중복되었다. 실험 1에서 ShallWeTalk 시스템을 사용했던 실험자가 학습효과 때문에 실험 2에서 BBS 시스템을 이용하면서 중복의견이 줄어들었다고 해석 된다.

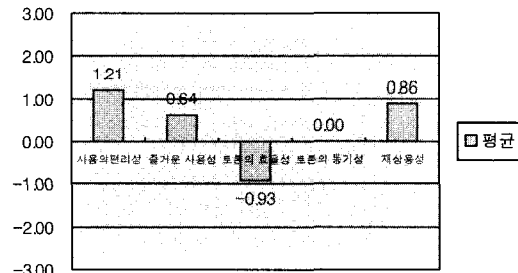
4.2.2 설문 결과 분석

참여자에게 사용성과 효과성을 테스트하기 위해 두 개의 질문서가 주어졌다. 각 질문은 7점 척도로 만들어 졌다. 즉, BBS 시스템이 ShallWeTalk 시스템보다 좋다고 생각되면 3에 가깝게 체크하고 반대라면 -3에 가깝게 체크하게 하였다. 16개의 피드백 중 14개가 적합하였다.

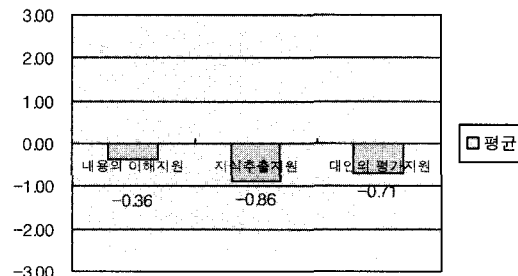
첫 번째 설문지는 Karacapilidis의 연구[9]에서 이용한 사용의 편리성, 즐거운 사용성, 토론의 효율성, 토론의 동기성, 그리고 재 사용성 등에 대한 사용성에 대한 질문을 포함하였다. <그림 5>에 첫 번째 설문의 결과를 보였다. 각 값은 질문에 대한 14개의 피드백에 대한 평균값을 의미한다. BBS 시스템이 세 개의 항목(사용의 편리성, 즐거운 사용성, 재 사용성)에서 좋게 나왔다. 아마도 사용자들이 BBS 시스템을 오래 사용하여 왔고 또 친숙하기 때문인 것 같다. 그러나, ShallWeTalk 시스템이 토론의 효율성 항목에서 더 좋게 나왔는데 대화 구조의 룰에 따라 대화가 이루어지기 때문인 것으로 해석된다. 두 시스템 모두 토론의 동기성에서는 같

다고 나왔다.

두 번째 설문지는 두 시스템의 내용의 이해지원, 지식의 추출 지원, 대안들에 대한 평가의 지원 등과 같은 효과성에 관한 질문을 포함하였다. <그림 6>에 두 번째 설문의 결과를 보였다. 세 항목 모두 ShallWeTalk 시스템이 BBS 시스템보다 좋다고 나왔다.



<그림 5> 사용성



<그림 6> 효과성

오픈 질문에서는 많은 참여자가 다자간 대화를 위한 새로운 구조 생성을 좀 더 쉽게 하고 입력 화면의 사이즈 조정을 가능하게 하는 기능을 요구하였다. 또한, 대화 구조와 대화 요소들을 이해하기 쉽도록 시스템 사용 시에 자세한 설명을 제공하는

기능을 요구하였다.

종합적으로, 실험 1과 실험 2에 관한 설문 결과를 볼 때, 현재의 프로토타입 시스템이 개선할 사항이 있긴 하지만, BBS 시스템보다 협업 방식의 문제 해결을 더 잘 지원한다는 결론을 내릴 수 있다.

## 5. 결 론

오늘날 우리가 맞닥뜨리는 문제들의 복잡성은 개인의 지적 능력을 초월하며, 이 점이 그러한 복잡한 문제들을 해결하기 위해 협업 접근법을 이용하게 한다. 그래서, 채팅 시스템이나 스크립트 기반의 시스템, 구조적 시스템 등과 같은 협업 지원을 위해 컴퓨터를 이용하는 많은 시도가 있었다. 그러나, 앞에서 밝힌 것처럼 이런 인터넷의 기본 서비스가 협업 기반의 문제 해결 프로세스를 만족할만하게 지원하지 않았다.

협업 방식의 다자간 대화지원 시스템 개발을 위해 전통적인 과학적 지식 획득 프로세스가 하나의 협업 방식의 문제 해결 프로세스라는 것을 기반으로 IBIS 모델을 선택하고 수정하였다. 수정된 프로세스는 솔루션 찾기와 솔루션 평가의 두 부 프로세스로 이루어 졌다. 그리고, 이 부 프로세스들을 위한 대화 구조로 의사소통행위 요소들을 이용하여 세트를 정의하였다. IBIS 모델과 비교하여 만든 구조는 앞장의 실험 결과에서 보인 것처럼 다자간 대화를 쉽게 이해 할 수 있을 만큼 충분히 간단하다.

본 논문의 기여는 다음과 같이 요약 될 수 있다. 첫째, 비 동기 대화에서 대화 내용의 섞임이 일어나지 않고 협업 방식의 문제 해결 프로세스를 지원하는 간단한 다자간 대화 구조를 제시하였다. 둘째, 본 연구에서 제시한 ShallWeTalk 시스템은 어떠한 다자간 대화 환경에서도 사용 될 수 있도록 다자간 대화 구조가 그 목적에 맞게 동적으로 정의 될 수 있게 지원한다. 셋째, ShallWeTalk 시스템은 사용자 사이에 빠른 협업을 지원하기 위해 이메일을 이용하여 자동으로 사용자에게 피드백을 보내

준다. 이 외에도 ShallWeTalk 시스템은 대안 솔루션들을 쉽게 비교하는 기능과 시스템 사용자의 기여를 계산하는 기능 등의 많은 기능을 제공한다.

그러나, 본 시스템은 좀 더 개선할 점을 가지고 있다. 각 대안 솔루션들 간에 순위와 그 순위를 표현 할 수 있는 기능이 있으면 좋을 것이다. 협업 방식의 문제 해결의 최종 단계에서는 제안된 솔루션 중에서 최선의 솔루션을 선택하는 의사결정이 이루어져야 하기 때문이다. 그리고, 유용성 테스트에서 좀 더 많은 주제에 대한 테스트가 필요하다. ShallWeTalk 시스템과 앞에서 언급한 기존 시스템들과의 비교도 이루어지면 좋을 것이다.

## 참 고 문 헌

- [1] Baker, M.J. and Lund, K., "Promoting Reflective Interactions in a Computer-Supported Collaborative Learning Environment," *Journal of Computer Assisted Learning*, 13(1997), pp.175-193.
- [2] Bunt, H.C., "Dialogue Control Functions and Interaction Design," *Proceedings of the NATO Advanced Research Workshop on Natural Dialogue and Interactive Student Modeling*, (1995), pp.197-214.
- [3] Conklin, J. and Begeman, M., "gIBIS : A Hypertext Tool for Exploratory Policy Discussion," *Proceedings of Computer-Supported Cooperative Work*, (1988), pp.140-152.
- [4] Cook, J. and Oliver, M., "Designing a Toolkit to Support Dialogue in Learning," *Computers & Education*, 38(2002), pp.151-164.
- [5] Cook, J., "Bridging the Gap between Empirical Data on Open-ended Tutorial Interactions and Computational Models," *International Journal of Artificial Intelligence in Education*, 12(2001), pp.85-99.

- [6] Cook, J., "The Role of Dialogue in Computer-based Learning and Observing Learning : an Evolutionary Approach to Theory," *Journal of Interactive Media in Education*, 2002(5).
- [7] Farnham, S., Chesley, H. R., McGhee, D. E., Kawal, R. and Landau, J., "Structured Online Interaction : Improving the Decision-Making of Small Discussion Groups," *Proceedings of Computer-Supported Cooperative Work*, (2000), pp.299-308.
- [8] Herring, S., "Interactional Coherence in CMC," *Proceedings of the Thirty-Second Hawaii International Conference on System Sciences*, 1999.
- [9] Karacapilidis, N., Papadias D., "Computer Supported Argumentation and Collaborative Decision Making : The HERMES System," *Information Systems*, Vol.26, No.4 (2001), pp.259-277.
- [10] Pfister, H-R. and Mühlpfordt, Martin, "Supporting Discourse in a Synchronous Learning Environment : The Learning Protocol Approach," *Computer Support for Collaborative Learning (CSCL)*, Boulder, CO, (Jan. 2002), pp.581-589.
- [11] Popper, K.R., *The Logic of Scientific Discovery*, Harper and Row, New York, 1968.
- [12] Rein G.L. and Ellis, C.A., "rIBIS : A Real-Time Group Hypertext System," MCC Technical Report STP-095-90, 1990.
- [13] Rittel, H. and Webber M., "Dilemmas in a General Theory of Planning," *Policy Sciences*, Vol.4(1973), pp.155-169.
- [14] Schrage, Michael, *Shared Minds : The New Technologies of Collaboration*, Random House, New York, 1990.
- [15] Smith, M. and Cadiz, J. and Burkhalter, B., "Conversation Trees and Threaded Chats," *Proceedings of Computer-Supported Cooperative Work*, (2000), pp.97-105.
- [16] Stahl, G., "Reflections on WebGuide : Seven Issues for the Next Generation of Collaborative Knowledge-Building Environments," *Proceedings of the Computer Support for Collaborative Learning (CSCL) Conference*, Palo Alto, CA, (Dec. 1999), pp. 600-610.
- [17] Stahl, G., "A Model of Collaborative Knowledge-Building," *Proceedings of the Fourth International Conference of the Learning Sciences*, (2000), pp.70-77.
- [18] Stahl, G., "WebGuide : Building Collaborative Learning on the Web with Perspectives," *Journal of Interactive Media in Education*, 2001(1).
- [19] Winograd, T., "A Language/Action Perspective on the Design of Cooperative Work," *Human-Computer Interaction*, 3(1988), pp. 3-30.