Journal of the Korean Institute of Industrial Engineers
Vol. 30, No. 1, pp. 1-10, March 2004.

CPFR 구현을 위한 다중 에이전트 시스템 구조 설계

김창욱[1†] · 김선일[1] · 윤정욱[1] · 박윤선[2]

[1]연세대학교 컴퓨터산업공학부 / [2]명지대학교 산업시스템공학부

Design of a Multi-Agent System Architecture for Implementing CPFR

Chang-Ouk Kim[1] · Sun-Il Kim[1] · Jung-Wook Yoon[1] · Yunsun Park[2]

[1]School of Computer & Industrial Engineering, Yonsei University, Seoul, 120-749
[2]Department of Industrial Systems Engineering, Myongji University, Yongin, 449-728

Advance in Internet technology has changed traditional production planning and control methods. In particular, collaborations between participants in supply chains are being increasingly addressed in industry for enhancing chain-wide productivity. A representative paradigm that emphasizes collaboration in production planning and control is CPFR(Collaborative Planning, Forecasting and Replenishment). In this paper, we present a multi-agent system architecture that supports the collaborations specified in CPFR. The multi-agent system architecture consists of event manager, data view agent, business rule agent, and collaboration agent. The collaboration agent systematically controls negotiation between supplier and buyer with the aid of collaboration protocol and blackboard. The multi-agent system has been implemented with EJB(Enterprise Java Beans).

***Keywords*:** CPFR, multi-agent, business rule, collaboration, blackboard, speech-act theory, EJB

1. 서 론

현재 공급망 운영에 대한 연구는 기존의 생산관리 방식에 지대한 변화를 가져오고 있다. 기존의 생산관리 방식은 개별 기업의 생산성 향상에 목표를 두고 있는 것에 반해서 공급망 이론은 개별 기업이 아니라 공급망 상의 모든 참여 기업들이 혜택을 볼 수 있는 것에 목표를 두고 있다. 이를 위해서 현재까지 많은 수리적 분석 방법이 제안되었다.

그러나 현실적으로 공급망 전체의 효율적인 운영을 위해서는 참여 기업 간에 예외적인 생산관리 상황을 협업(Collaboration)을 통하여 해결할 수 있는 시스템이 필요하다(Johnson, 1998). 협업 시스템은 인터넷망을 통해 빠르게 전달되는 소비 시장의 변화를 적시에 참여 기업들의 생산 관리에 반영할 수 있다는 장점이 있다. 최근 이러한 요구로 기존의 생산관리 패러다임을 탈피하는 새로운 생산관리 모형으로서 VICS(Voluntary Interindustry Commerce Standards)에서 제시한 CPFR(Collaborative Planning, Forecasting, and Replenishment) 모형이 있다. CPFR은 공급망 상의 참여 기업들이 효율적인 정보 공유와 예외적인 생산관리 상황을 협업을 통하여 해결하여 참여 기업의 효율성을 극대화하는 방법 중의 하나이다.

본 논문에서는 CPFR 환경하에서 예외 사항을 해결하기 위한 협업 시스템 구조를 제시한다. 예외 사항이란 참여 기업 간에 공동으로 수립된 계획에서 벗어나 새로운 협업을 통해서 해결해야 할 생산관리 문제를 의미한다. 이를 위해서 첫째로, 협업 시스템을 지원하기 위한 다중 에이전트 시스템(Multi- Agent System) 구조를 제시한다. 둘째로, 이 구조 내에서 예외 사항들에 대한 참여 기업 간 협업을 효율적으로 지원하는 방법으로 블랙보드 기반 협업 시스템(Blackboard-Based Collaboration System) 구조를 제시한다. 여기서 블랙보드는 컴퓨터 내의 가상 협업 공간을 의미한다(Ito and Salleh, 2000). 또한 화행 이론(Speech-Act Theory)을 기반으로 하는 협업 프로토콜(Collaboration Protocol)을 제시한다. 화행 이론은 수행

본 연구는 한국학술진흥재단(KRF-2002-003-D00448)의 지원에 의하여 수행되었음.
† 연락저자 : 김창욱 교수, 120-749 서울시 서대문구 신촌동 134번지 연세대학교 컴퓨터산업공학부, Fax : 02-730-8430,
E-mail : kimco@yonsei.ac.kr
2003년 7월 접수; 2003년 10월 수정본 접수; 2003년 12월 게재 확정.

문(Performatives)이라고 해서 특별한 문장으로 이를 발화(Activation)하여 어떤 일을 행하고, 수행문의 적정 조건이 충족되었는지의 여부에 따라 적정 혹은 부적정으로 평가하는 것을 말한다(Winograd and Flores, 1988). 블랙보드 기반 협업 시스템은 협업 프로토콜을 이용하여 예외 사항들을 참여 기업들이 효율적으로 처리할 수 있는 기반을 마련해 준다.

본 논문의 구성은 다음과 같다. 제2장에서는 본 연구에서 다루는 전체적인 CPFR 프로세스에 대한 설명을 한다. 제3장에서는 CPFR을 구현하기 위한 다중 에이전트 시스템 구조, 블랙보드 기반 협업 시스템 및 협업 프로토콜을 설명한다. 제4장에서는 다중 에이전트 시스템과 블랙보드 기반 협업 시스템 구조를 구현하기 위해 사용된 EJB(Enterprise Java Beans) 기술에 대하여 설명한다. 마지막으로 제5장에서는 본 연구의 결론을 맺고 추후 연구 과제에 대해서 설명한다.

2. CPFR 프로세스

CPFR 모형(<Figure 1> 참조)은 기본적으로 9단계의 프로세스들로 이루어지며, 각 프로세스들은 거래업체들 간의 정보 공유와 협상에 의해서 진행된다. 우선 CPFR 모형의 프로세스들을 살펴보면 다음과 같다.

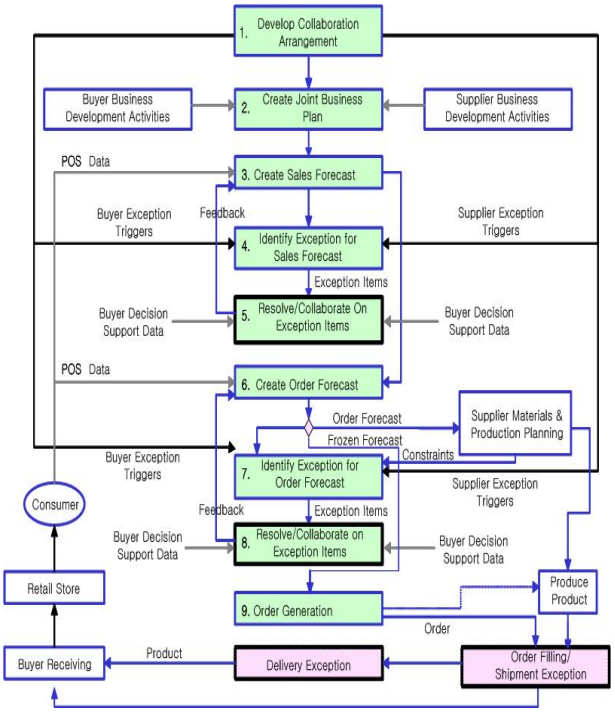

Figure 1. CPFR process model.

- 1단계: 협업에 대한 합의서 작성
  - 구매업체와 공급업체는 각사의 사업 목표를 가지고 협업 합의서를 구체화한다.
- 2단계: 공동 사업 계획
  - 양사는 사업 목표에 따른 협업의 전략이나 전술 등의 범위를 결정한다.
- 3단계: 판매 예측 작성
  - 양사는 기본 공유 데이터를 가지고 판매 예측을 작성한다.
- 4단계: 판매 예측 예외 확인
  - 변경된 판매 예측 데이터가 발생하면 작성된 판매 예측과 비교하며 예외 사항이 발생하는지 분석한다.
- 5단계: 판매 예측 예외 사항 협의
  - 발생한 판매 예측 예외 사항들에 대한 양사의 데이터들을 가지고 예외사항을 해결한다.
- 6단계: 주문 예측 작성
  - 판매 예측 데이터와 양사의 기본 공유 데이터를 가지고 주문 예측을 작성한다.
- 7단계: 주문 예측 예외 확인
  - 변경된 주문 예측 데이터가 발생하면 작성된 주문 예측과 비교하며 예외 사항이 발생하는지 분석한다.
- 8단계: 주문 예측 예외 사항 협의
  - 발생한 주문 예측 예외 사항들에 대한 양사의 데이터들을 가지고 예외 사항을 해결한다.
- 9단계: 주문 실행
  - 주문 예측에 따른 주문을 시행하고 주문 정보는 양사의 생산 및 판매 시스템으로 전송된다.

2단계에서 협업의 전략이나 전술 등의 범위 결정은 예외 사항을 명시하는 것을 의미한다. 한 예로 '기간별 판매 예측이 10% 상하로 변경되면 협업을 통하여 판매 예측을 다시 한다.'라는 것이 있을 수 있으며, 예외 사항 결정은 전적으로 공급업체와 구매업체가 1-2단계를 거쳐서 합의해야 한다. 6단계의 주문 예측은 양사 간 합의된 판매 예측을 기반으로 기간별 주문량을 정하는 것을 의미한다. CPFR 모형의 핵심은 예외 사항에 대한 협업을 진행하는 4-5단계와 7-8단계에 있는데, 본 연구는 이 단계들을 공급업체와 구매업체가 인터넷 기반 정보 시스템을 이용하여 효율적으로 진행시킬 수 있는 방법을 제시한다.

3. CPFR을 구현하기 위한 다중 에이전트 시스템 구조

3.1 다중 에이전트 시스템 구조 및 구성 요소

에이전트는 자율성과 지능적인 요소를 가지고 사용자를 대신 해서 일을 처리해 주는 역할을 한다(Choi, 1997). 하지만 이런 에이전트들도 사용자가 정한 일이나 경험에 의해 알아낸 목적을 단독으로 달성하지 못하는 경우가 많다. 단독의 에

이전트가 가질 수 있는 경험은 한정되어 있으나 사용자의 요구는 다양할 뿐만 아니라, 예측을 할 수 없기 때문이다. 그러나 다중 에이전트 시스템은 에이전트의 자율성과 지능적인 요소를 가지고 특정 목적에 대하여 사용자가 원하는 작업을 자동적으로 해결해 주는 특징을 가지고 있다. 즉, 사용자의 복잡한 요구는 하나의 에이전트에서 처리하기 어려운 경우가 많기 때문에 여러 에이전트끼리 정보 공유를 통해서 문제를 해결하려는 협업적 다중 에이전트 시스템이 필요한 것이다(Yang and Choi, 1999).

본 논문에서도 CPFR의 개념을 포괄하고 단독 에이전트의 단점을 보안한 다중 에이전트 시스템 구조 모형을 <Figure 2>와 같이 제시한다. 다중 에이전트 시스템은 여러 개의 요소들로 구성되며, 각 부분은 다음과 같은 역할을 한다.

- **사건 관리자(Event Manager):** 공급업체와 구매업체에서 요청하는 정보 조회 사건과 CFPR 프로세스의 4-5단계와 7-8단계를 수행하게 하는 각종 예외 사항(예: 공급업체 재고 부족, 구매업체의 제품 가격 상승과 제품 판촉 행사 일정 통보 등)을 관리한다. 공급업체와 구매업체는 각 사의 사용자 인터페이스를 통하여 사건 관리자에게 사건을 전달하며, 사건 관리자는 사건의 종류에 따라서 정보 조회 사건은 데이터 뷰 에이전트에 전달하고, 예외 사항은 비즈니스 규칙 에이전트에게 전달한다.
- **데이터 뷰 에이전트(Data View Agent):** 공급 업체와 구매업체는 이 에이전트를 통하여 협업 참여자들의 정보를 조회할 수 있다. 이 에이전트는 메타 데이터(Meta Data)를 참조하여 허용된 공유 범위 안에서 상대방의 정보를 검색할 수 있게 해 준다.
- **메타 데이터(Meta Data):** 공급업체와 구매업체가 합의한 정보의 공유 범위를 기록하고 있는 데이터베이스이다. 이 범위에 어긋나는 검색에 대해서는 데이터 뷰 에이전트를 통하여 정보 열람 승인 거부 결과 값을 전달한다.
- **비즈니스 규칙 에이전트(Business Rule Agent):** 사건-조건-활동 기반 비즈니스 규칙 데이터(Event-Condition-Action Based Business Rule Data)에 입력되어 있는 사건-조건-활동 정보와 비교하여 예외 사항 조건에 부합되는지를 조사한다(CPFR 4 또는 7단계 프로세스). 예외 사항 조건에 부합되면 협업 에이전트(Collaboration Agent)에게 예외 사항을 통보한다.
- **사건-조건-활동 기반 비즈니스 규칙 데이터(Event-Condition-Action Based Business Rule Data):** 협의된 예외 사항을 사건-조건-활동 식으로 기록하여 보관한다(Goh *et al.*, 2001). 일반적으로 사건이란 공급업체와 구매업체의 데이터 중에서 변경 값이 CPFR 프로세스 4 또는 7단계를 실행하게 만드는 것을 의미한다. 조건이란 데이터의 변경 값이 일정 범위를 넘어서야만 예외 사항으로 인정하는 부분이다. 활동은 사건-조건이 모두 만족하면 발생하는 것으로, 본 연구에서는 공급업체와 구매업체 간의 블랙보드 기반 협업을 의미한다. 예를 들어 구매업체의 제품 가격이 상승하여 이를 가격 데이터에 반영하면 사건이 발생하며, 만일 조건이 '가격 상승률이 10%이상'이라 하면 이 조건이 만족되었을 때만 블랙보드를 생성하여 협업을 실행시킨다.
- **협업 에이전트(Collaboration Agent):** 블랙보드 기반 협업 시스템을 제어하는 역할을 한다(<Figure 3> 참조). 구체적으로 협업 에이전트는 (1) 양사의 협업 결과를 CPFR 거래 데이터에 기록하고, (2) 양사의 협업 진행 내용을 과거 협

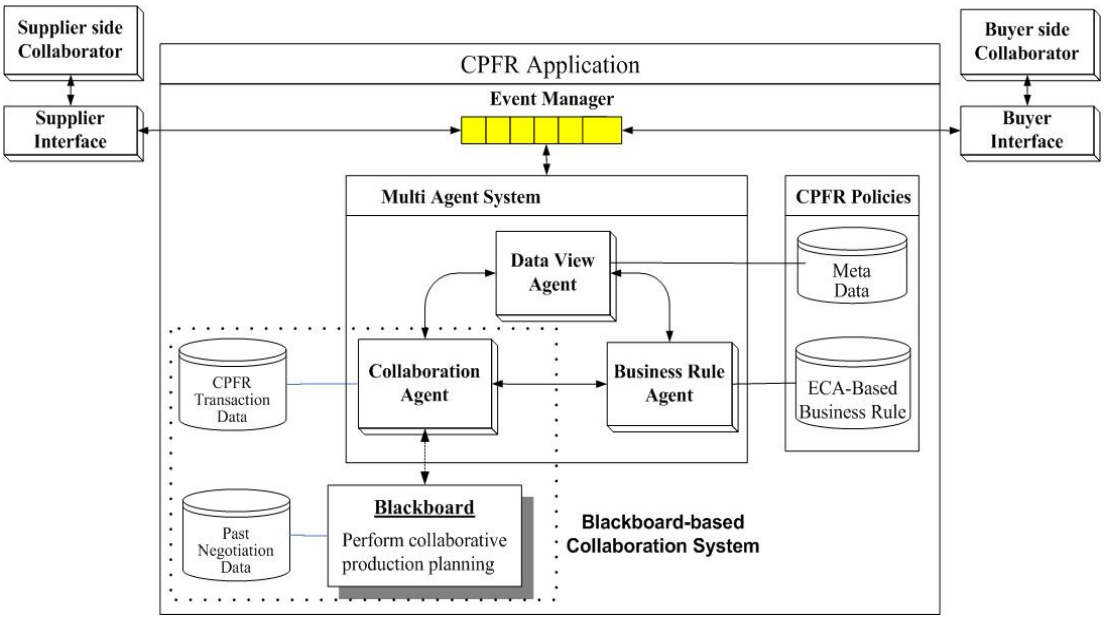

Figure 2. Multi-agent system architecture for implementing CPFR.

상 데이터베이스에 기록하며, (3) 협상을 위한 블랙보드를 제어하며, (4) 비즈니스 규칙 에이전트로부터 예외 사항이 넘어오면 CPFR 거래 데이터로부터 예외사항과 유사한 과거의 협상 결과들을 아젠다 메커니즘(Agenda Mechanism)을 이용하여 양사에 제공한다. 만일 협업 참여자들이 제공된 과거 협상 결과를 수용하지 않으면 협업 에이전트는 블랙보드를 생성하여 협업을 시작한다. 따라서 협업 에이전트는 블랙보드의 생성 및 소멸, 협상에서 최종 합의된 결과 등을 협업 참여자들에게 전송한다.

- **아젠다 메커니즘(Agenda Mechanism):** 협업 에이전트가 공급업체와 구매업체에게 과거의 예외 처리 해결 방안을 제시할 때, 어떤 우선순위로 어떤 해결안을 제시해야 할지를 결정한다(Sadeh *et al.*, 1998). 본 논문에서 제시하고 있는 아젠다 메커니즘은 예전과 같은 사건이 생겼을 때, 이 사건에 해당하는 사건-조건과 일치하는 과거 해결안들을 찾는다. 그리고 찾은 해결안들은 최근 해결안을 최우선순위로 공급업체와 구매업체에게 제시한다.
- **블랙보드(Blackboard):** 일종의 협상 테이블로 협업 에이전트에서 제시한 해결안을 공급업체와 구매업체가 동의하지 않을 경우에 생성된다. 이곳은 협상 권한을 가진 협업 참여자들이 서로의 해결안을 제시하고 참조하며, 서로의 의견을 충분히 조율할 수 있는 협상의 장소로 쓰인다. 이때 각 협업 참여자들의 협상 조정은 화행 이론을 이용한 협업 프로토콜을 따른다.
- **블랙보드 기반 협업 시스템(Blackboard Based Collaboration System):** <Figure 3>에서 보듯이 CPFR 다중 에이전트 시스템 중에서 협업을 위한 핵심 부분을 의미한다.

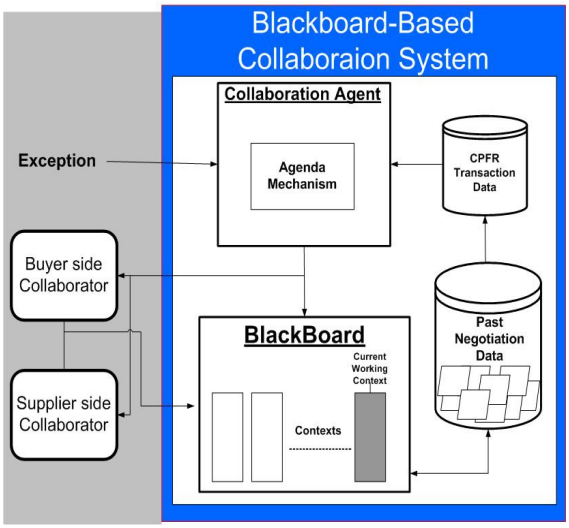

Figure 3. Blackboard-based collaboration system.

- **CPFR 거래 데이터(CPFR Transaction Data):** 협업 에이전트에서 처리한 협상 결과 데이터를 저장하고 있으며, 이 데이터베이스는 CPFR 시스템을 위한 지식 자원이 된다. 이 데이터에 대한 전체적인 협상 내용이 있는 데이터는 과거 협상 데이터가 가지고 있다.
- **과거 협상 데이터(Past Negotiation Data):** 블랙보드 상에서 진행된 모든 내용들에 대한 기록을 저장하며, 블랙보드의 지식 자원이 된다.
- **협업자(Collaborator):** 협의되지 않은 예외 사항에 대한 해결안을 블랙보드 상에서 협상하는 양 업체의 협업 참여자를 의미한다.

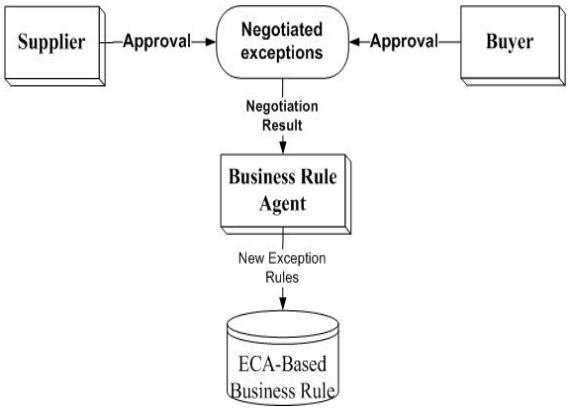

Figure 4. Establish CPFR polices data flow.

CPFR 프로세스 중 1, 2단계에서 합의된 예외 사항들은 <Figure 4>에서 보듯이 공급업체와 구매업체가 합의하면 비즈니스 규칙 에이전트에 의해서 사건-조건-활동 식으로 표현되어 CPFR 정책으로 저장되게 된다.

3.2 블랙보드 기반 협업 시스템

블랙보드 기반 협업 시스템을 설명하기 전에 먼저 사건-조건-활동 규칙의 표현에 대해서 설명하기로 한다. <Figure 5>는 주문에 대한 예외 사항을 사건-조건-활동 규칙으로 표현한 예이다. 첫 번째 예는 구매업체가 합의된 주문 예측량 10000보다 큰 주문량을 요구했을 때 발생하는 예외 사항을 기술한 것이고, 두 번째 예는 공급업체의 제품 가격이 변화했을 때 발생하는 주문의 재 예측 요구를 표현한 것이다. 합당한 예외 사항을 찾기 위해서 구매업체와 공급업체의 협업 참여자들은 사용자 인터페이스를 통해서 <데이터 종류, 변경값>으로 구성된 사건을 사건 관리자에게 전달해야 한다. <Figure 5>의 첫 번째 예를 들면 사건은 <주문량, 12000> 등이 될 것이다.

앞에서도 간단하게 서술하였듯이 블랙보드 기반 협업 시스템은 <Figure 1>에서 제4단계와 제7단계에서 발생하는 예외 사항들을 효율적으로 해결하기 위한 시스템이며, <협업 에이전트, 블랙보드, CPFR 거래 데이터, 과거 협상 데이터>로 구성되어 있다. 사건 관리자로부터 사건이 비즈니스 규칙 에이전트에게 전달되면 이 에이전트는 합당한 예외 사항을

찾아서 협업 에이전트에게 전달한다. 만일 비즈니스 규칙 에이전트에게 전달된 사건이 조건에 맞는 게 없으면 예외 사항이 발생하지 않은 것으로 간주하여 협업 에이전트를 호출하지 않는다. <Figure 6>은 예외 사항이 비즈니스 규칙 에이전트로부터 넘어 올 경우, 협업 에이전트가 아젠다 메커니즘에 따라 CPFR 거래 데이터에서 유사한 사항의 과거 협상 결과의 요약 데이터를 공급업체와 구매업체에게 제시한 예를 보여 준다.

Purchase Order Rule Base:

Example1 (Order quantity)
Event Purchase order
Condition order quantity per an order >10000
Action Invoke collaboration agent

Example2 (Price)
Event Purchase order
Condition price per unit <>100
Action Invoke collaboration agent

Figure 5. Event-condition-action rule base.

협업 에이전트는 제시된 과거 협상 해결 방안 데이터들을 공급업체와 구매업체에게 보여주며, 이 데이터로 예외 처리를 할 것인지 질의한다. 협업 참여자들이 이 데이터에 모두 동의하면 블랙보드를 생성할 필요 없이 모든 세션이 끝나고 예외 사항이 해결되게 된다. 그러나 한 쪽이라도 동의하지 않는다면(<Figure 6>의 Reject 버튼) 협업 에이전트는 블랙보드를 생성하게 되고 참여 기업 간 협업적 협상을 할 수 있도록 한다. 협업 참여자들은 이 예외 사항에 대한 협상안을 블랙보드에 올리기 시작하고, 이 협상안이 만족될 때까지 협상은 계속된다.

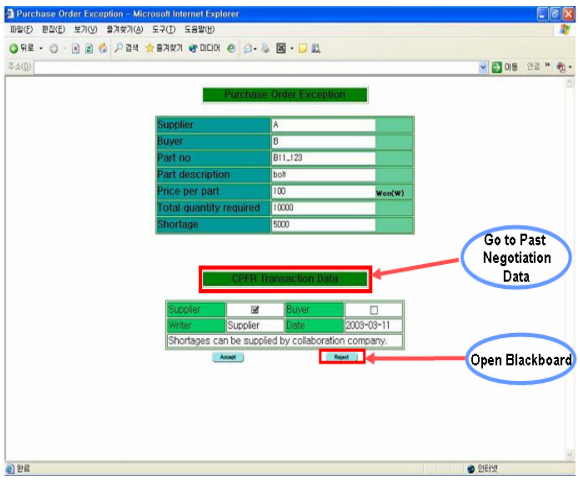

Figure 6. A layout of CPFR transaction data.

이때의 블랙보드(<Figure 7> 참조)는 게시판 형식으로 공급업체와 구매업체의 협업 참여자들에게 보여 진다. 협상은 서로의 목적을 만족시킬 경우에 종료된다. 이 때 블랙보드에서 협상된 데이터는 과거 협상 데이터에 저장되며, 이 협상 결과 최종 승인된 데이터는 분리되어 CPFR 거래 데이터에 저장된다.

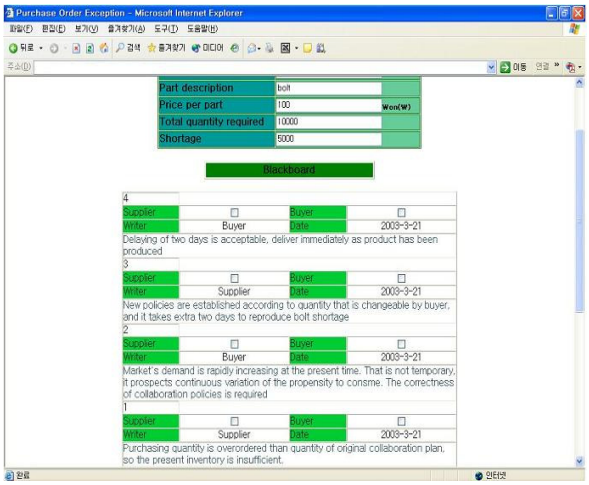

Figure 7. A layout of blackboard.

앞에서도 언급하였듯이 협업을 체계적으로 진행하기 위해서는 협업 프로토콜이 필요하다. 협업 프로토콜이란 협업 절차와 틀을 체계적으로 만든 것을 의미한다(Huhn and Stephen, 2001). 블랙보드 기반 협업 시스템에서는 공급업체와 구매업체의 협업 참여자가 예외 처리 해결 방안을 찾는 복잡한 협상 과정을 거치게 되는데, 이를 고정적인 협상 순서로 표현하기에는 어려운 점이 많다. 따라서 협업 참여자들은 유동적인 협상 진행 과정을 가져야 한다. 이를 위해서 본 연구에서는 1988년에 Winograd와 Flores가 기존의 화행 이론을 형식화(Formalization)해서 제안한 도메인-독립 상태 조직도(Domain-Independent State Machine)를 협업 프로토콜을 표현 하는 데 적용시키고자 한다(Winograd and Flores, 1988). 이 조직도는 일의 진행 상태를 중심으로 유동적이고 반복적인 협상을 가능하게 하는 장점이 있다(Colombetti and Verdicchio, 2002; Stefano and Santoro, 2002). 그리고 협업 프로토콜을 통해서 협업 참여자들이 서로의 의견을 충분히 조율할 수 있는 가상의 협업 공간을 제공해 주어야 하는데 바로 그 역할을 게시판 형식의 블랙보드가 한다.

본 논문의 협업 프로토콜을 설명하기 전에 필요한 기호와 용어들을 정의하면 다음과 같다.

: 대화 가능 상태

: 종료 상태

: 화행(Speech Act)

S: 공급업체(Supplier)
B: 구매업체(Buyer)
CA: 협업 에이전트(Collaboration Agent)
BB: 블랙보드(Blackboard)

- **Request:** 화자(Speaker)가 어떤 행동(Action)을 하기 전에 청자(Hearer)에게 의견을 물어보는 행위를 말한다.
- **Commit:** 화자가 제안한 행동을 실행(Execute)하기 전에 청자가 제안을 수락하여 행위를 약속하는 행위를 말한다.
- **Propose:** 청자가 거부(Reject)한 행위에 대해서 화자가 청자에게 새로운 행위를 제시한다.
- **Withdraw:** 화자와 청자가 더 이상 대화하기를 원하지 않을 때에 대화를 중단하는 것이다.
- **Reject:** 화자가 청자에게 제안 한 행위에 대해서 청자가 거부하는 행위를 말한다. 그러나 Withdraw와는 달리 대화를 중단하는 것은 아니다.
- **Execute:** 화자가 청자에게 허락(Commit)을 받아 행위를 실행하는 것을 의미한다.

위에서 정의된 S, B, CA 및 BB는 행위 주체로서 서로 대화를 주고받는 화자와 청자의 역할을 한다. 그리고 행위 주체끼리의 화행은 진술, 요청, 약속을 하는 식의 언표내적 행위(Illocutionary Act)를 한다. 이때 사용되는 용어들은 기존의 화행 이론을 바탕으로 블랙보드 기반 협업 시스템 상에서의 협업 프로토콜에 맞게 변형시킨 것이다.

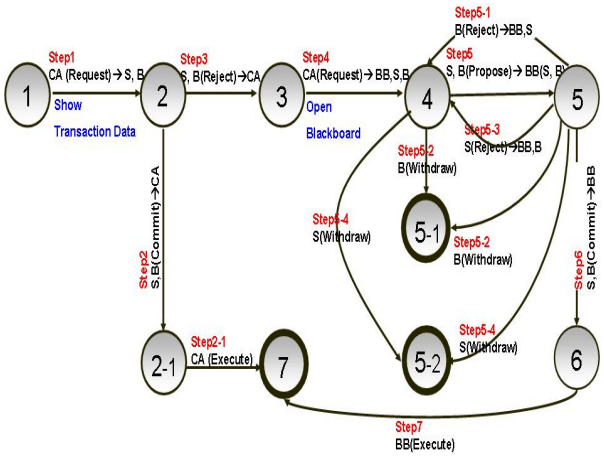

Figure 8. A collaboration protocol in blackboard-based collaboration system.

블랙보드 기반 협업 시스템의 협업 프로토콜은 <Figure 8>과 같이 다이어그램으로 표현할 수 있다. 이 협업 프로토콜에서 원안의 숫자는 협업 과정의 상태를 1에서부터 7까지 나타낸 것이다. 구체적인 예로 <Figure 5>의 Example1(식 (1))을 가지고 <Figure 8>과 같이 Step1부터 Step7까지 협업 프로토콜의 진행 과정을 살펴보기로 한다.

Event Purchase order
Condition order quantity per an order>10000 (1)
Action Invoke collaboration agent

이 때 Step1부터 Step7까지 각 Step에 적용되는 형식은 아래의 식 (2)와 같이 일반화시킬 수 있으며, 이 식은 각 Step의 진행 사항을 나타낸다.

Speaker (Speech-Act) → Hearer (2)

Step 1

CA(Request) → S, B: 구매업체가 합의된 주문 예측량 10,000보다 많은 주문량 12,000을 요구했을 경우는 협업 참여자들끼리 정한 주문 예측량보다 높기 때문에 예외 사항이 발생한다. 그러면 협업 에이전트는 아젠다 메커니즘에 따라 예외 처리 해결안을 CPFR 거래 데이터로부터 가져온다. 그리고 협업 에이전트는 '부족한 수량은 다른 협력업체로부터 공급받을 수 있다'라는 식의 기존 해결안을 공급업체와 구매업체에게 보여주면서 이 해결안을 수락할지의 여부를 물어본다.

Step1은 협업 에이전트(CA)가 구매업체(B)와 공급업체(S) 모두에게 의견을 물어보는(Request) 행위이다. 양사 모두 협업 에이전트의 해결안을 수락(Commit)하는 행위는 Step2에서 일어난다. 그리고 한 쪽이라도 거부(Reject)하는 행동은 Step3에서 일어난다. 하지만 포기(Withdraw)는 Step4가 될 때까지는 일어나지 않는다. 그 이유는 블랙보드가 열리고 나서 공급업체와 구매업체가 협상을 진행할 때만 포기를 할 수 있게 프로토콜을 제시하였기 때문이다.

Step 2

S, B(Commit) → CA: 구매업체와 공급업체 모두 협업 에이전트가 제시하는 기존의 해결안에 동의한다. 이 때 부족한 수량 2,000은 다른 협력업체가 구매업체에게 공급하게 된다.

Step2는 공급업체(S)와 구매업체(B)가 협업 에이전트(CA)가 제시한 해결안을 수락(Commit)함으로써 행위를 약속하는 행위이다.

Step 2-1

CA (Execute): 구매업체와 공급업체 간에 결정된 기존의 해결안을 실행한다. 즉, 구매업체는 다른 협력업체에게 부족한 수량을 주문한다.

협업 에이전트(CA)가 수락된 기존의 해결안을 실행(Execute)하는 행위이다.

Step2는 공급업체와 구매업체 모두 수락(Commit)을 약속하는 단계이기 때문에 이미 합의가 끝난 것이다. 즉, 이 단계에서 양사 중에 한쪽이 거절(Reject)하거나 포기(Withdraw)하는 행위를 할 수 없다. 또 양사 중에 한쪽이 새로운 해결안을 제시(Propose)하는 행위도 할 수 없는 것이다.

Step 3

S, B(Reject) → CA: Step2에서 구매업체나 공급업체는 협업 에이전트가 제시한 해결안을 거부한다. 즉, 구매업체나 공급업체는 다른 협력업체가 부족한 수량을 공급하거나 다른 협력업체로부터 공급받기를 원하지 않는다. 하지만 구매업체나 공급업체는 서로 간의 협상을 거부하거나 그만둔 것은 아니다.

Step2에서 공급업체(S)와 구매업체(B)가 협업 에이전트(CA)가 제시한 해결안을 거절(Reject)하는 행위이다. 하지만 만약에 한쪽에서 포기(Withdraw)를 원할 경우는 블랙보드가 열린 후에만 가능하다.

Step 4

CA(Request) → BB, S, B: 협업 에이전트는 Step 3에서 거부한 기존의 해결안 대신 새로운 해결안을 찾기 위해서 예외 사항과 함께 블랙보드를 생성한다. 공급업체와 구매업체는 블랙보드를 통해서 서로의 의견을 주고받을 수 있는 협상 테이블로서 블랙보드를 사용한다.

협업 에이전트(CA)는 공급업체(S)와 구매업체(B)에게 블랙보드(BB)를 통해서 새로운 의견을 물어보는(Request) 행위를 한다.

Step 5

S, B (Propose) → BB(S, B): 구매업체와 공급업체는 자신들이 원하는 해결안을 블랙보드에 올린다. 예를 들어 공급업체는 '부족 수량은 재생산하는 데 3일의 추가 시간이 소요됨'식의 문구를, 구매업체는' 2일의 지연 시간은 수용 가능하다. 하지만 3일의 지연시간은 안 된다.'식의 문구를 올린다. 이렇게 협업 참여자들은 블랙보드를 통해서 서로의 합의점을 찾을 때까지 협상을 계속한다.

공급업체(S), 구매업체(B)가 블랙보드(BB)를 통해서 공급업체(S)나 구매업체(B)에게 제안(Propose)하는 행위이다. 이때 구매업체가 새로운 해결안에 대해서 거부(Reject)할 때는 Step5-1에서, 공급업체가 거부할 때는 Step5-3에서 행위가 일어난다. 하지만 양사 모두 수락(Commit)한다면 Step6에서 수락 행위가 일어난다.

Step 5-1

B(Reject) → BB, S: 구매업체는 블랙보드를 통해서 공급업체가 제시한 3일의 추가시간을 거절한다. 그러면 공급업체는 블랙보드를 통해서 구매업체가 거부한 사실을 알게 된다. 그리고 다시 양사는 서로가 만족할 수 있는 해결안을 찾기 위해서 협상을 계속한다.

구매업체(B)가 블랙보드(BB)를 통해서 공급업체(S)가 제시한 해결안을 거절(Reject)하는 행위이다. 하지만 협상은 계속된다. 그리고 구매업체가 협상을 포기(Withdraw)할 때만 Step5-2에서 포기 행위가 일어난다.

Step 5-2

B(Withdraw): Step5-1에서 구매업체는 공급업체가 제시한 3일의 추가시간을 거부하고 협상을 중지한다. 이것은 더 이상의 협상을 원하지 않는다는 뜻이다.

구매업체(B)가 협상을 포기(Withdraw)하는 행위이다. 포기 행위 이후에는 제안(Propose)을 할 수 없다.

Step5-3

S(Reject) → BB, B: 공급업체는 블랙보드를 통해서 구매업체가 제시한 2일의 지연시간을 수용할 수 없음을 나타낸다. 그러면 구매업체는 블랙보드를 통해서 공급업체가 거부한 사실을 알게 된다. 그리고 다시 양사는 서로의 해결안을 찾기 위해서 협상을 계속한다.

공급업체(S)가 블랙보드(BB)를 통해서 구매업체(B)가 제시한 해결안을 거절(Reject)하는 행위이다. 하지만 협상은 계속된다. 그리고 구매업체가 협상을 포기(Withdraw)할 때만 Step5-4에서 포기 행위가 일어난다.

Step 5-4

S(Withdraw): Step5-3에서 공급업체는 구매업체가 제시한 2일의 지연시간이 수용 가능하지 않다고 판단하고 협상을 중지한다. 이것은 구매업체와 더 이상의 협상을 원하지 않는다는 뜻이다.

공급업체(S)가 협상을 포기(Withdraw)하는 행위이다. 포기

행위 다음에는 제안(Propose)을 할 수 없다.

Step 6

> **S, B(Commit) → BB:** Step5에서 구매업체와 공급업체는 수차례 지연 시간에 대한 해결안을 블랙보드 상에서 주고받는다. 그리고 양사는 만족할 수 있는 해결안을 찾고 협의를 하게 된다. 이때 모든 협상 기록은 블랙보드 상에 남게 된다.

공급업체(S), 구매업체(B)가 블랙보드(BB)를 통한 협상 끝에 해결안을 찾고 양사가 수락(Commit)하는 행위이다.

Step 7

> **BB(Execute):** Step6에서 합의된 지연 시간이 CPFR 거래 데이터로 가서 저장되며 이와 유사한 예외 상황이 발생할 경우, CPFR 거래 데이터에서 협업 에이전트가 참조할 수 있게 된다.

블랙보드(BB) 상에 수락된 해결안을 실행(Execute)하는 행위이다.

위에서 살펴본 Step1~7까지의 예는 팩스(fax), 전화, 또는 회의를 이용하는 기존의 협업 방법과 다르게 협업이 웹 상에서 프로토콜을 이용하여 통제 가능하다는 것을 보여준다. 따라서 본 연구의 의의는 시간과 공간의 제한을 극복할 수 있는 웹의 장점을 이용하여 복잡한 협업 과정을 효율적으로 통제하도록 만들었다는 데 있다. 하지만 일-대-일의 참여업체간 협업을 가능케 하는 협업 프로토콜을 나타내기보다는 다-대-다 관계에서의 협업을 지원하는 프로토콜이 필요하며 이를 위해서는 추가적인 연구가 필요하다.

4. EJB를 이용한 구현 모형

본 논문에서는 다중 에이전트 시스템 구조를 EJB를 이용하여 구현하였다. EJB를 이용하면 3-Tier 혹은 n-Tier를 지원하는 엔터프라이즈급의 분산 객체 시스템을 구현할 수 있다. CPFR 가이드라인(Johnson, 1998)은 공유, 중앙 서버, 분산 서버 등의 모형을 제시하고 있다. EJB는 세 가지 모형 중 분산 서버 형태에 적합한 방식이다. EJB로 구현한 CPFR 다중 에이전트 시스템은 다음과 같은 특징을 가지게 된다.

CPFR정책을 따르는 유사한 도메인에는 거의 변경 없이 기존 컴포넌트를 적용하여 원하는 서비스를 제공할 수 있다.

연관이 적은 도메인일지라도 본 연구에서 제안한 구조를 따른다면 시스템을 재개발할 필요 없이 필요한 컴포넌트를 새로 추가하거나 컴포넌트의 일부분을 수정함으로써, 개발 기간, 자금 및 인력 등을 절감할 수 있다.

웹(Web)을 지원하므로 시스템 구축 시 추가 비용이 적으며 ERP, CORBA 등의 시스템과 연결하여 사용할 수 있다.

SUN(Sun Microsystems, Inc.)에서 제공하는 풍부한 API (Application Programming Interface)를 사용할 수 있으며 보안,

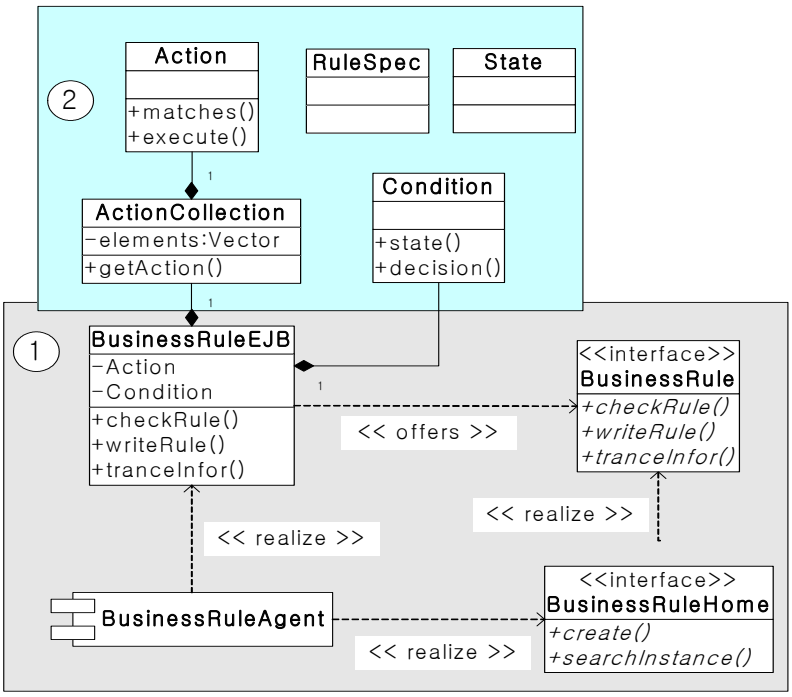

Figure 9. Inside of a business rule agent.

트랜잭션, 객체 간 통신 등의 기술이 내부적으로 지원되므로 업무 규칙을 집중적으로 구현할 수 있다.

EJB의 n-Tier 구조는 계층마다 상호 연관성이 줄어 유지 보수 비용이 줄어들고 계층 간 통신이 가능하며 동일한 코드를 재사용함으로써 업무 표준화가 이루어진다.

이러한 시스템의 높은 유연성 때문에 참여업체들은 추가적 비용이나 시스템 개발 기간에 따르는 부담 없이 쉽게 CPFR 다중 에이전트 시스템으로 전환할 수 있게 된다. <Figure 9>은 다중 에이전트 시스템 구조에서 여러 컴포넌트들 중 하나인 비즈니스 규칙 에이전트의 내부 구조이다.

<Figure 9>의 컴포넌트는 열거, 대행 등의 디자인 패턴을 충족시키도록 설계하여 재사용이 가능하도록 하였다(Gamma *et al*., 1995). EJB 명세에 적합한 컴포넌트들은 기본적으로 ①처럼 두 개의 인터페이스와 하나의 구현 클래스가 필요하다(Enterpirse blue print). BusinessRuleHome 인터페이스는 Business RuleEJB 클래스의 객체 생성과 참조를 담당한다.

②의 클래스들을 포함하고 있는 Business RuleEJB 클래스는 내부적으로 실행되며 요청한 일을 수행할 수 있는 메소드를 가지고 있다. ②는 State, Action, Condition, Action Collection, RuleSpec. 클래스를 가지고 있다.

RuleSpec 클래스는 사건-조건-활동 기반 비즈니스 규칙 데이터의 내용을 처리하며, State 클래스는 사용자 정보, 제품 정보, 예측 정보 등의 데이터를 포함한다. Condition 클래스는 사용자의 사건을 판단하고 Action 클래스는 비즈니스 규칙에 따라 수행될 작업을 나타낸다. ActionCollection은 Action 클래스를 담고 있으며, Condition과 Action을 대응시키는 역할을 한다.

비즈니스 규칙 에이전트 개체는 사용자가 Business Rule 인터페이스의 메소드를 호출함으로써 작업이 시작된다. 메소드가 호출되면 Business RuleEJB 클래스의 실제 구현된 메소드안에서 Action Collection 클래스와 Codition 클래스가 호출된다. Condition 클래스는 state(), decision() 메소드를 가지며 RuleSpec, State클래스를 참조하여 어떤 일을 수행해야 하는지 판단한다. ActionCollection 클래스는 Condition 클래스의 판단 결과를 가지고 해당하는 Action을 한다.

<Figure 10>은 본 논문의 다중 에이전트 시스템 구조를 컴포넌트를 이용하여 구현한 컴포넌트 다이어그램이다. Web/Client 컴포넌트 사용자가 시스템에 접속했을 때 보여 지는 부분이며 이를 통해 작업이 시작된다. Login 컴포넌트는 사용자에 따른 권한을 할당하는 컴포넌트이다.

③의 SaleForcasting(판매 예측), OrderForecasting(주문 예측), ForecastingUpdate(예측 갱신), Order(주문 실행) 컴포넌트는 다중 에이전트 시스템에 사건을 전달하는 화면 컴포넌트이다.

④의 사건 관리자 컴포넌트는 ③에서 전달된 사건을 정의된 진행 순서에 따라 각 에이전트 컴포넌트로 전달한다. 데이터 뷰 관리자 컴포넌트는 다른 참여업체들의 정보를 조회할 수 있게 해 주며 메타 데이터를 참조하는 Emeta Data 컴포넌

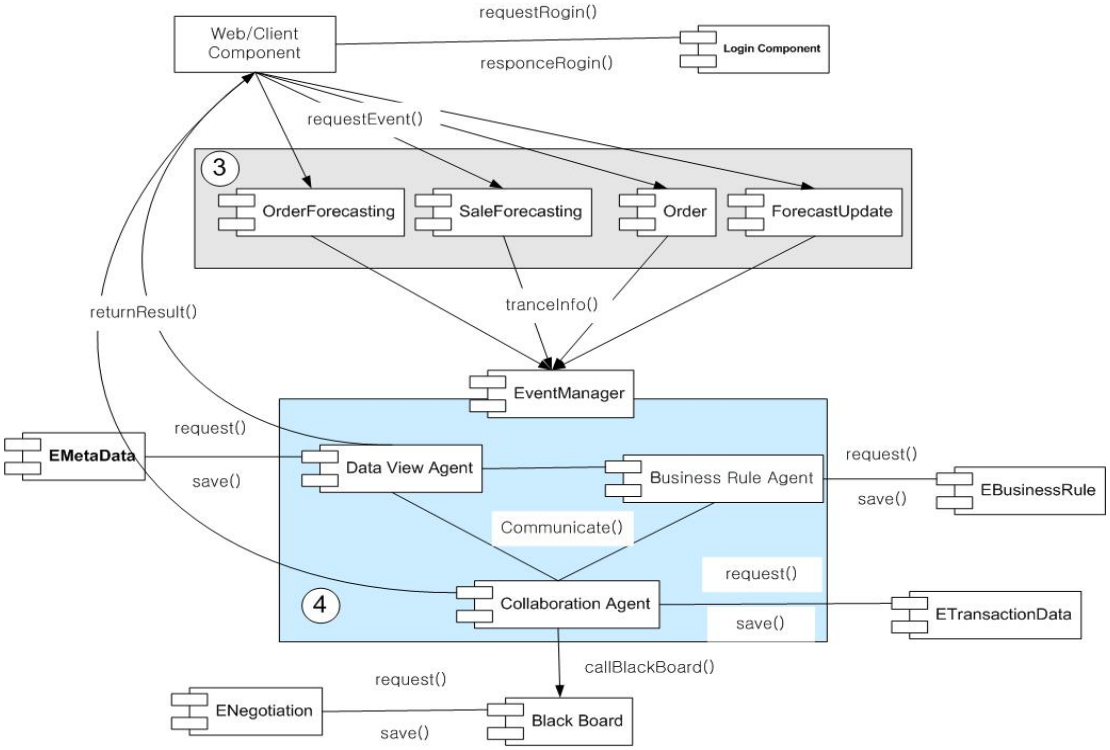

Figure 10. UML component diagram for multi-agent.

트를 통해 허용된 공유 범위 안에서 상대방의 정보를 검색할 수 있으며 다른 컴포넌트에게 이 정보를 전달할 수 있다. 비즈니스 규칙 에이전트는 사건-조선-활동 기반 비즈니스 규칙 데이터를 참조하는 Etransaction Data를 이용하여 적합한 업무 규칙을 선택하여 협업 에이전트에게 전달한다.

협업 에이전트 컴포넌트는 양사의 협업 내용을 CPFR 거래 데이터를 관리하는 ETransaction Data 컴포넌트를 통하여 기록하고, 협상 결과를 Web/Client 컴포넌트에게 전달한다.

5. 결 론

본 논문에서는 기업 간 협업 시스템의 대표적인 사례인 CPFR의 구현을 위해서 다중 에이전트 기반 정보 시스템 구조를 설계하였고, CPFR내에서 발생하는 예외 사항들을 해결하기 위한 방법으로 블랙보드 기반 협업 에이전트 구조와 화행 이론을 이용한 협업 프로토콜을 제시하여 유동적 협업 관계를 나타내었다. 또한 이들을 EJB를 이용한 웹 기반 애플리케이션으로 구현하였다.

추후 연구 과제로는 첫째로, 다중 에이전트 시스템을 두 업체 이상의 협상이 가능한 시스템으로 확장하는 것이다. 둘째로 기업들이 가지고 있는 다양한 형태의 데이터들을 서로 공유할 수 있도록 XML(eXtensible Markup Language) 기반 Ontology Transfer System(Riza and Oguz, 2002)에 대한 연구가 진행되어야 할 것이다. 셋째로 개발된 소프트웨어를 많이 수정하지 않고도 기존의 다양한 시스템 환경하에서 사용할 수 있도록 재사용성이나 호환성에 대한 연구도 함께 진행되어야 한다. 마지막으로, 블랙보드 상에서의 효율적인 협상이 진행되도록 자동화된 협상 알고리즘에 관한 연구가 진행되어야 할 것이다.

참고문헌

Choi, J. M. (1997), Introduciton of agent, *The Institute of Electronics Enginners of Korea,* **15**(3), 7-16.

Colombetti, M. and Verdicchio, M. (2002), An Analysis of Agent Speech Acts as Institutional Actions, *Proceedings of The International Joint Conference on Autonomous Agents and Multiagent Systems*, 1157-1164.

Enterprise Blue Prints, Available at http://java.sun.com/blueprints.

Gamma, E., Helm, R., Johnson, R. and Vlissides, J. (1995), *Design Patterns: Elements of Reusable Object-Oriented Software*, Addison-Wesley.

Goh, A., Koh, Y.-K., and Domazet, D.S. (2001), ECA-Based Support for Workflows, *Artificial intelligence in engineering*, **15**(1), 37-46.

Huhn, M. N., Stephen, L. M. (2001), Automating Supply Chains, *IEEE Internet Computing,* **5**(4), 90-93.

Ito, T and Salleh, M. R. (2000), A BlackboardBased Negotiation for Collaborative Supply Chain System, *Journal of Materials Processing Technology,* **107**(1/3), 398-403.

Johnson, M. (1998), Collaboration Data Modeling: CPFR Implementation Guidelines, *White Paper*, Available at http://www.cpfr.org.

Papazoglou, M. P. (2001), Agent-Oriented Technology in Support of E-Business, *Communications of ACM*, **44**(4), 71-77.

Riza, C. E. and Oguz, D. (2002), A Multi-Agent System Infrastructure for Software Compenent Market-Place: An Ontological Perspective, *SIGMOD record*, **31**(1), 55-60.

Sacile, R., Paolucci, M. and Boccalatte, A. (2000), The Make-IT PROJECT: Manufacturing Agents in a Knowledge-based Environment driven by Internet Technologies, *Proceedings. Academia/Industry Working Conference on*., 281-291

Sadeh, N. M., Hildum, D. W., Laliberty, T. J., McA'Nulty, J., Kjenstad, D., and Tseng, A. (1998), A blackboard architecture for integrating process planning and production scheduling, *Concurrent Engineering: Research and Applications*, **6**(2).

Tan, G. W. and Shaw, M. J. (1998), Applying Component Technology to Improve Global Supply Chain Network Management, *Proceedings of The International Confefence On Information Systems*, **19**, 296-301

Winograd, T. and Flores, F. (1988), *Understanding Computers and Cognition*, Addison-Wesley, U.S.A.

Yang, J. Y. and Choi, J. M. (1999), Collaborative Agent Systems, *The Institute of Electronics Enginners of Korea,* **26**(1), 25-33.

Yonghui, F.R.P., Rajesh, P., Robert, D. S. and Jingru W. (2000), Multi-Agent Enabled Modeling And Simulation towards Collaborative Inventory Management Supply Chains, *Proc. of the 2000 Winter Simulation Conference.*, 1763-1771.