

# 실시간 공급사슬 환경하에서의 거래자동화 에이전트 개발

박영재<sup>1\*</sup>

<sup>1</sup>동명대학교 경영대학 경영정보학과

## Development of Automated Trading Agent in Real-Time Supply Chain Environment

Young Jae Park<sup>1\*</sup>

<sup>1</sup>Department of Management Information Systems, Tongmyung University

**요약** 자재구매계획, 생산계획, 그리고 고객에게 상품을 전달하기 위한 유통계획 등 공급원재료 및 자재를 공급하는 공급자로부터 고객에게 제품을 전달하기까지의 전 과정을 포함하는 공급사슬관리는 과거 폐쇄적이고 정적인 것에서 개방적이고 동적인 환경으로 변화되고 있다. 이러한 공급사슬의 환경적 변화는 기업에게 공급사슬의 운영과 관리 방식 있어 다른 접근방법을 요구하고 있다. 그러나 현재까지는 이러한 환경변화에 대응하기 위한 실험이나 연구는 부족한 상황이다. 한편 TAC-SCM (Trading Agent Competition Supply Chain Management)는 실험적인 환경을 제공하고 있다. TAC-SCM 환경은 공급자-생산자-고객의 3단계 사슬로 구성된다. 생산자는 PC 조립업체로 에이전트로 구현된다. 각 에이전트는 고객으로부터의 주문을 성사시키기 위해 서로 경쟁해야 하며, PC를 조립하기 위해 필요한 부품을 구매하기 위해서도 서로 경쟁해야 한다. 이를 위해 각 에이전트는 고객과의 협상, 생산계획, 자재구매계획, 상품인도를 위한 유통계획 등의 일반적인 공급사슬관리 업무를 자동으로 수행해야 한다. 본 연구에서는 2008년도 TAC-SCM 게임 규칙을 따른 에이전트를 구현하였으며 구현된 알고리즘과 실험결과를 소개하였다.

**Abstract** Recently, business environment is spreading from singular business to supply-chain problem. SCM(Supply Chain Management) is changing from closed and static to opened and dynamic environment, so these environment changes of SCM require to enterprise different approach than past. As supply chain is becoming opened and dynamic, a claim is being naming that converse of real-time the get reaction from to recognizing the level of supply chain problem, grope of alternative plan to solving this problem and choosing optimum alternative plan raise the enterprise competitive. But until present, the research to deal with these changes of SCM is on insufficient situation. Thus in this paper proposes the agent system by a way of realizing dynamic supply chain. The agent is going to accomplish the management activities in real-time depends on environment and given condition and designed for solving problems development and maintenance. Also, algorithms performance of the agent is tested on TAC SCM environment. Agents in TAC SCM are simulations of small manufacturers, who must compete with each other for both supplies and customers, and manage inventories and production facilities.

**Key Words** : SCM; Agent; Automated Negotiation; Automated Transaction

### 1. 서론

공급사슬관리는 원재료에서부터 완제품의 배달에 이

르기까지 공급사슬 전반에 걸친 활동들을 계획하고 조정하는 경영활동이다. 오늘날의 공급사슬은 관련된 기업들의 수가 점점 더 많아질 뿐만 아니라 경영의 글로벌화에

이 논문은 2008년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음 (KRF-2008-B00095).

\*교신저자 : 박영재(yjpark@tu.ac.kr)

접수일 10년 08월 26일

수정일 10년 10월 06일

게재확정일 10년 11월 19일

따라 공급사슬도 글로벌화 되고 있다. 이러한 글로벌 경영환경에서의 효과적인 공급사슬관리는 변화하는 시장의 수요에 시기적절하고 비용효과적인 방법으로 대응할 수 있는 기업의 능력에 직접적으로 영향을 미치기 때문에 기업 경쟁력의 핵심이다. 현재까지의 공급사슬관계는 핵심적인 거래파트너와의 장기계약에 따른 정적인 관계가 일반적이며, 다양한 형태의 기업들이 이종의 환경에서 관련 기업에 속한 관리자 즉, 사람에 의해서 계약의 성립과 계획들이 이루어지고 관련 정보만이 관리되었다. 그러나 시장의 변화에 따라 공급자와 고객 간의 보다 나은 공급사슬을 구축하기 위해서는 보다 유연하고 동적인 환경에서의 실증적인 연구가 필요하다. 공급사슬은 연간 수백조 달러의 시장을 형성하고 있기 때문에 공급사슬관리 능력의 개선으로 인한 잠재적 효과는 엄청난 것이다. 그러나 이를 위한 실증적인 연구를 수행하기에는 매우 어려운데 그 이유는 복잡한 공급사슬의 형태와 동적인 거래 환경을 효과적으로 지원하기 위한 연구 모형의 수립이 어렵다는 점[5]과 실제 환경에 적용했을 경우 만약의 실패에 따른 높은 비용의 부담 때문이다[4].

한편 고객의 요구사항이 다양화되고 경쟁이 가속화되면서 경영환경의 불연속성이 증대하고 있다. 예측 가능한 환경에서 효율성을 추구했던 기업들은 역동적인 시장 상황에서 기업이 당면한 위기와 기회를 신속하게 인지하고 대응하는 능력이 요구되고 있다. 변화에 유연하게 대응하는 민첩한 조직(agile organization)과 고객 요구사항에 신속하게 대응하는 감지-반응 기업(adaptive enterprise)에 대한 논의는 이전부터 제기되어왔으며 최근에는 실시간 기업(RTE; Real-Time Enterprise)이라는 개념으로 확산되고 있다[9]. 이는 기업의 하부조직에서부터 최고 의사결정자에 이르기까지 모든 정보와 지식이 실시간으로 공유되는 경영환경을 구축하기 위한 방법론이다. 또한 단순히 내부업무뿐만 아니라 파트너와 고객사의 업무 즉 공급사슬 전체를 실시간화 함으로써 경영환경 변화에 공급사슬이 즉각적으로 대응하는 것을 말한다. 즉, 현재의 경영환경은 기업의 주요 업무 프로세스에서 발생하는 정보를 필요한 적재적소에 실시간으로 전달하여 즉각적인 모니터링과 신속한 대응이 가능한 기업역량을 요구하는 것이다. 오늘날 기업이 직면하고 있는 현실세계는 M&A, 제휴 등을 통한 글로벌 경쟁, 정보기술 중심의 급속한 기술 발달로 경영환경이 급변하고 있으며, 시장에서의 고객의 힘은 점점 강화되고 있을 뿐만 아니라 그 요구사항들도 다양해지고 있다. 이러한 경영환경 속에서 기업들은 얼마나 빨리 환경변화를 인지하고 고객의 요구사항에 신속하게 대응할 수 있느냐가 바로 기업 경쟁력 강화의 요체로 인식되고 있다. 이와 같이 RTE의 핵심개념은 상황인식,

적절한 대안마련, 적절한 대처(실행)이라는 위기관리 및 대처 사이클의 속도를 실시간화하여 기업의 경쟁력을 극대화하는 것이라 하겠다[6,7].

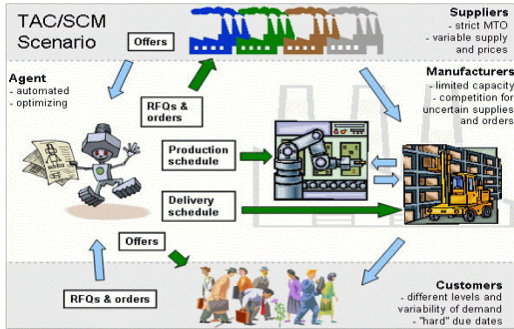
이와 같이 RTE란 공급사슬상에서의 종단간(end-to-end) 프로세스들의 사이클을 최대한 단축하여 실시간화하는 개념으로 이를 위해서는 상황인식과, 대안 분석 및 최적대안 제시, 그리고 상황에 반응하기 위한 에이전트와 같은 지능형 기술의 필요성이 강조되고 있다. 예로 근자에 들어 공급사슬관리의 전 단계를 자동화하는 에이전트 기술의 발전에 학계와 산업계 양쪽 모두 많은 관심을 보이고 있다[1, 2, 3]. 그러나 공급사슬관리와 관련된 일들을 지능적으로 수행하기 위한 자동화된 에이전트는 불확실성, 자원의 최적사용, 그리고 변화하는 시장 환경에 적응하기 위한 계획들을 세울 수 있어야만 한다. 이를 위해서는 에이전트가 감당해야 할 과업의 수가 너무 많기 때문에 공급사슬의 전 과정을 다루기 위한 에이전트의 구현 역시 매우 어려운 일이다. 또한 에이전트 기반 공급사슬관리 연구의 어려운 점은 자동화된 전략들을 실제 기업에서 적용하기에는 매우 어렵다는 것이다.

그러나 TAC-SCM(Trading Agent Competition Supply Chain Management)은 간단한 게임 규칙이지만 동적 공급사슬에 많은 참여자들이 충분한 경쟁 속에서 다양한 변화를 실증적으로 실험할 수 있는 기회를 제공하기 위해 설계되었다. TAC-SCM은 자동화된 전략을 지닌 독립적인 에이전트를 구현하여 다른 에이전트들과 경쟁하는 다양한 모의실험을 통해 공급사슬관리를 위한 에이전트를 연구할 수 있는 유일한 테스트베드를 제공하고 있다. 따라서 본 연구에서는 TAC SCM 게임 규칙에 따른 에이전트를 구축하여 실험을 하였다. 이렇게 함으로써 실제 적용시 실패에 따른 비용적 손실에 대한 위험을 줄일 수 있고, 특히 공급사슬의 신속한 대응에 대한 효과를 실증적 실험을 통해 검증해 볼 수 있다.

이와 같이 본 연구에서는 급변하는 경영환경에 대응하기 위한 실시간 공급사슬관리 에이전트를 개발 및 실험하였다.

## 2. TAC SCM

본 장에서는 TAC SCM 게임 시나리오와 규칙에 관한 개요를 설명하고자 한다. 보다 자세한 사항은 TAC SCM 사이트의 공식 게임 규칙서[5]를 참조할 수 있다.



[그림 1] TAC-SCM 시나리오

TAC-SCM은 그림 1과 같이 고객-제조업체(PC 조립업자)-공급자(PC 부품 제조 및 공급자)의 3단계의 공급사슬 환경으로 구성되어 있다. 6개의 제조업체들이 게임에 동시에 참여할 수 있으며 에이전트로 구현된다.

시나리오는 다음과 같다. 고객은 제조업체 즉, 에이전트들에게 구매하고자 하는 PC의 유형, 수량, 구매가격, 납기일의 내용이 담긴 RFQ를 보낸다. RFQ를 받은 에이전트들은 고객으로부터 받은 RFQ중 원하는 RFQ에 대해 판매가격을 제시하여 고객에게 입찰한다. 이때 에이전트는 고객이 제시한 PC 유형, 수량, 납기일은 반드시 만족시켜야 하며 고객이 제시한 구매가격을 초과하는 판매가격을 제시할 수 없다. 고객은 6개의 에이전트들이 제시한 Offer들 중 최소가격을 제시한 에이전트에게 Order를 낸다. 6개의 에이전트들은 고객과 주문을 성사시키기 위해서 서로 경쟁을 해야만 한다.

한편 에이전트가 고객으로부터 받은 주문의 PC를 생산하기 위한 부품을 공급받기 위해서는 해당 부품들을 제조하는 공급업체로부터 부품을 조달해야 한다. CPU, 마더보드, 메모리, 디스크 드라이브를 공급하는 8개의 공급업체로 구성되어 있다. 에이전트들은 해당 부품을 조달하기 위해서 서로 경쟁을 해야만 한다. 부품이 조달되면 에이전트는 부품들을 조립하여 고객이 원하는 납기일 내에 해당 제품을 고객에게 배달하여야 한다. 만약 납기일을 초과하면 가격의 일정비율만큼 벌금이 부과되며 납기일에서 5일을 초과하면 고객의 주문은 자동 취소된다.

TAC SCM에서는 15초를 하루로 220일 동안 진행된다. 따라서 에이전트는 15초 안에 고객의 RFQ들 중에 입찰할 RFQ를 정해서 가격을 제시해야 하며, 에이전트 및 공급업체의 일일 생산능력은 한정되어 있기 때문에 게임에서 승리하기 위한 부품 조달계획, 일정계획 그리고 배달계획 등의 전략과 알고리즘을 게임 환경에 맞게 효과적으로 수립해야 한다.

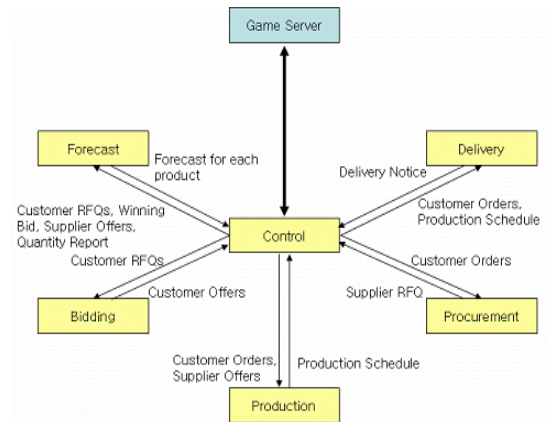
에이전트가 판매한 금액은 게임 서버의 은행에 예금되

며 부품 구매비용, 재고유지비용, 그리고 납기일을 초과한 벌금 등이 은행의 잔고에서 차감된다. 6개의 에이전트들 중 게임이 끝났을 때 은행의 잔고가 가장 많은 에이전트가 최종 승자가 된다. TAC-SCM은 참가한 에이전트들끼리 조를 나누어 예선과 본선을 거쳐 최종 6개의 에이전트가 결승에서 우승을 겨룬다. 결승전은 매년 저명한 국제학술대회에서 치러지고 있다.

### 3. DAUTAC

TAC SCM에서는 다양한 전략을 실험 해 볼 수 있다. 예를 들면 생산 방식으로는 재고기반생산방식(Make-To-Stock) 혹은 주문기반생산방식(Make-To-Order) 등을 고려할 수 있다. 생산전략을 어떻게 수립하느냐에 따라 이후의 연관된 프로세스들의 전략도 달라진다. 재고기반생산방식을 위해서는 수요예측이 기반이 되며 이에 따라 BOM(Bill of Material)에 따른 MRP(Material Requirement Plan)를 전개하고 필요한 부품을 구매하기 위한 구매전략을 수립해야 한다. 그러나 주문기반으로 할 경우 주문된 제품을 구매하기 위한 일정계획과 부품구매 전략이 수립된다. 이와 같이 어떠한 전략을 수립하는가에 따라 관련된 프로세스들의 전략이 구체화 되어야 하며 각 프로세스 별로 예를 들면 수요예측 방법론, 일정계획 방법론들도 다양한 방법들이 고려될 수 있다. 본 연구에서는 재고기반생산방식에 따른 생산전략을 수립하였다.

#### 3.1 DAUTAC 구조



[그림 2] DAUTAC2008 구조

DAUTAC은 그림 2에서와 같이 6개의 모듈로 구성되어 있으며 처리 절차는 다음과 같다.

- 1) Forecast는 모듈은 단순회귀분석을 기반으로 사용자의 향후 수요를 예측한다.
- 2) 예측된 수요를 충족시키기 위해 필요한 부품을 Procurement 모듈이 구매한다.
- 3) 부품이 납입되면 납입된 부품정보를 바탕으로 Production 모듈이 일정계획을 수립하고 이를 바탕으로 ATP(Available-To-Promise) 정보를 생성한다.
- 4) Bidding 모듈은 Production 모듈의 ATP 정보를 기반으로 고객의 RFQ에 대응하여 Offer를 한다.
- 5) 고객으로부터 주문이 들어오면 Delivery 모듈은 납기일 내에 배달할 수 있도록 Delivery 일정을 수립한다.

### 3.2 Forecast

예측 모듈은 과거 고객의 수요 정보를 바탕으로 미래 수요를 예측하여 필요한 부품의 조달 및 생산계획을 세우기 위한 가장 중요한 정보를 제공하는 모듈이다. TAC-SCM에서의 고객의 수요, 즉  $d$ 날에 발행되는 RFQ(Request For Quote)의 수는 컴퓨터시장의 세가지 유형([표 2]에서의 시장유형)별로 아래의 수식과 같이 포아송 분포에 따르며 Trend( $\tau$ )는 -00.1에서 0.01사이의 변수를 발생시켜 특정일에 발행되는 RFQ의 수를 변화시키고 있다. 게임 첫째 날의 Q0는 컴퓨터시장의 세가지 유형([표 2]에서의 시장유형)에 따라 High와 Low Range에서는 [20, 100]에서 Middle Range에서는 [30, 120]의 범위에서 임의로 정해진다.

- $N = \text{poisson}(Qd)$
- $Qd+1 = \min(Q_{\max}, \max(Q_{\min}, \tau dQd))$
- $\tau d+1 = \max(\tau_{\min}, \min(\tau_{\max}, \tau d + \text{random}(-0:010:01))$

고객의 RFQ상의 납기일은 현재일에서 향후 3일부터 12일까지의 사이에서 임의로 정해지며 가격은 해당 PC를 조립하기 위해 필요한 부품의 Base Price의 합에서 75~125%의 범위에서 임의로 정해진다. 각 RFQ상에서의 주문수량은 [1,20]의 범위에서 역시 임의로 정해진다.

이와 같은 고객의 수요변화를 예측하기 위해서 DAUTAC2008의 수요예측은 최소자승법을 이용한 단순회귀분석을 사용하였다. 수요와 관련된 수식을 보면 계절적 요소나 변동적인 요소는 포함되지 않고 경향만을 포함하고 있기 때문에 경향을 분석하면 수요를 예측할 수 있다. 경향만을 가지는 수요예측에서는 지수평활법, 이동평균법, 회귀분석 등이 있다. 본 연구에서는 제한된 시간 내에 여러 가지 예측과 계획을 수립해야 하므로 단순회귀분석을 사용하였다. 입력데이터로는 과거 10일 동안의

고객 RFQ이며 향후 12일간의 고객 수요를 예측한다. 향후 12일간의 수요를 예측하는 이유는 고객이 발행하는 RFQ상의 Due date가 향후 3일부터 12까지의 범위를 가지기 때문이다.

### 3.3 Procurement

Forecast에서 예측된 수요정보를 바탕으로 부품조달계획을 수립한다. 예측된 수요는 전체시장의 수요이므로 전체 시장의 수요 중 자신의 에이전트가 생산하여 판매할 목표시장 및 수량을 결정해야 한다. 본 연구에서는 예측된 전체 수요를 6개 에이전트가 평등하게 기회를 가진다는 가정 하에 예측된 수요를 [4,6]의 범위에서 나눈 값을 목표시장 및 수량 값으로 사용하였다. TAC-SCM에서는 공급자의 생산능력이 한정되어 있기 때문에 에이전트가 필요한 부품을 원하는 시기에 원하는 수량을 공급받기 위해서는 경쟁을 벌여야 한다. 따라서 경쟁의 패배로 부품을 공급받지 못하면 입찰 또는 고객의 주문을 충족시킬 수 없게 된다. 필요한 부품을 조달하기 위해 에이전트가 공급자에게 RFQ를 보내면 공급자는 세 가지의 Offer 보내오게 된다. 첫 번째는 에이전트가 요구한 RFQ 상의 수량과 다른, 즉 요구수량보다 낮은 수량을 제시하거나, 두 번째는 요구한 납기일보다 빠른 납기일을 제시할 수 있다. 세 번째는 고객이 요구한 수량과 납기일을 모두 준수한 Offer이다. 세 번째의 경우는 문제 될 것이 없다. 그러나 첫 번째의 경우 요구수량보다 낮기 때문에 생산계획에 차질이 생길 수 있으며, 두 번째의 경우는 재고비용이 발생한다. 에이전트는 이러한 경우 Offer를 수락하여 주문을 할 것인지에 대한 의사결정이 필요하다. Procurement에서는 수요예측을 바탕으로 공급자에게 RFQ를 보내는데 현재일로부터 5일과 6일 후에 생산할 부품들에 대해서 작성한다. 공급자로부터 부품이 입고된 다음날에 그 부품을 이용해 생산할 수 있으므로 부품구매를 위한 RFQ상의 Due Date는 생산일보다 하루 이른 시간으로 정하며 가격은 Base Price  $\times 1.25$  의 값으로 고정하였다.

### 3.4 Production

Production에서는 Procurement와 같은 방식으로 정한 목표시장 및 수량을 생산하기 위한 일정계획을 수립한다. 에이전트의 일일 생산능력은 2,000cycle로 한정되어 있다. 본 연구에서의 일정계획방법은 배낭문제 모델링하여 생산순서만 결정하도록 하였다. 즉 조달된 부품을 바탕으로 수요가 높은 제품, 제품 사이클이 낮은 제품, 기본 가격이 높은 제품부터 배낭에 넣어서 생산하는 하도록 하였다. 생산일정계획이 수립되면 Production 모듈은 이

를 바탕으로 ATP 정보를 생성한다. 생산일정계획을 수립하여야 할 대상은 당일 배달되어온 부품들이다. 하루 만에 다 생산해야 하는데 하루 즉 구매한 부품들의 조립을 위해 필요한 사이클이 2,000 사이클 넘을 경우 다음날로 넘겨서 생산일정계획을 수립한다.

### 3.5 Bidding

Bidding 모듈은 고객의 RFQ에 대응하여 Offer를 해야 한다. 이때 고객이 제시한 납기일과 수량을 반드시 만족시켜야 한다. 따라서 Bidding 모듈은 Production 모듈에서 생성된 ATP 정보와 비교하여 고객이 제시한 수량과 납기일을 지킬 수 있는 RFQ에 한해서 Offer한다. 이때 어떤 RFQ부터 ATP와 비교해야 하는 문제가 있는데 다음 수식의 값이 가장 큰 RFQ부터 ATP 정보와 비교 및 할당하도록 하였다.

- 고객이 제시한 가격 + ((Due Date - 현재일) × ((Base Price × S × 보관기일) ÷ 수량)), 단 S = 게임 시작시 [25%, 50%]에서 게임서버에서 임의로 정함

Bidding 모듈이 결정해야 할 또 하나 중요한 내용은 가격결정이다. 고객이 제시한 가격 이내에서 가격을 입찰해야 한다. Bidding 모듈의 가격 결정식은 다음과 같다.

- Bid Price = Procurement Price × S × [0.6, 1.3], 단 S(재고비용) = 게임 시작시 [25%, 50%]에서 게임 서버에서 임의로 정함

마진율인 [0.6, 1.3]의 범위는 수요 변화와 목적인 승률의 달성 정도에 따라 변동되며 목표 승률 즉, 오픈한 RFQ 대비 주문의 비율은 80%로 주었다. ATP에 할당할 RFQ와 가격이 정해지면 해당 RFQ의 수량을 ATP에서 차감하여 ATP를 갱신한다. 더 이상 할당할 RFQ 및 ATP가 없으면 Bidding을 종료한다.

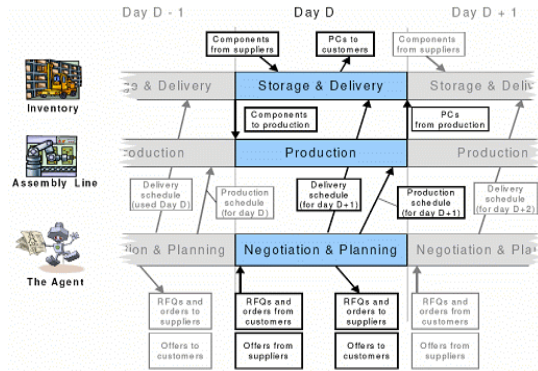
### 3.6 Delivery

Delivery 모듈은 생산되어 창고에 보관된 제품을 납기일 내에 고객에게 배달하기 위한 일정을 수립한다. 배달 일정계획이 수립되어 배달을 요청하면 배달은 그 다음날에 이루어지기 때문에 배달요청메시지는 납기일 하루 이전에 서버로 보내어 저장한다.

### 3.7 Control

Control 모듈은 TAC-SCM 서버와의 통신과 내부 모듈

간의 메시지를 관리한다. 메시지는 시기와 순서가 중요하다. 다음의 그림 3과 같이 TAC-SCM에서는 서버와 주고받는 메시지와 공급자와 주고받는 메시지는 시기와 순서가 있다.



[그림 3] Timing and Sequence of Message

Control 모듈은 각 모듈에서 생성된 메시지들의 시기와 순서를 고려하여 외부에 전달한다. 또한 각 모듈을 호출하여 필요한 정보와 메시지를 생성할 수 있도록 한다.

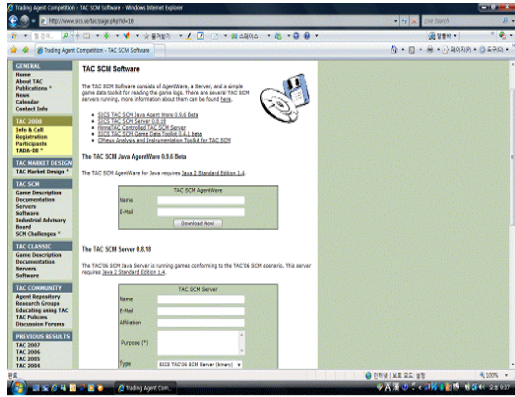
게임이 시작되면 Control 모듈은 고객의 RFQ와 주문을 접수하고 공급자에게 RFQ를 제시하고 공급자의 Offer에 대해 Order를 발행한다. 재고를 관리하고 Factory에 생산일정계획을 보낸다. 이를 위해 Control 모듈은 Forecast 모듈을 호출하여 고객의 수요를 예측하게 한다. Control 모듈은 재고정보와 부품구매정보 그리고 수요예측 정보를 Production 모듈에 보내고 이를 바탕으로 Production 모듈을 생산일정계획을 수립한다. 생산일정계획이 수립되면 Control 모듈을 이를 받아 서버로 전송한다. Control 모듈은 Bidding 모듈을 호출하여 고객으로 받은 RFQ 정보를 전송하고 이를 바탕으로 생성한 Offer를 넘겨받아 서버로 전송하여 Bidding이 이루어지도록 한다. Procurement 모듈을 호출하여 구매해야 할 부품에 대한 RFQ를 생성하도록 한다. 공급자로부터 Offer를 받으면 해당 Offer에 대해 모두 Order를 발송한다. 완성이 된 제품들에 대해서는 납기일에 제품이 배달될 수 있도록 Delivery schedule을 생성하여 서버에 전송한다. 게임이 다음날로 넘어 가기 전에 로그 데이터, 정보 그리고 메시지 큐에 들어 있는 데이터들을 청소한다.

## 4. 구현 및 실험환경

TAC-SCM을 위한 에이전트를 구현하기 위해서는 아

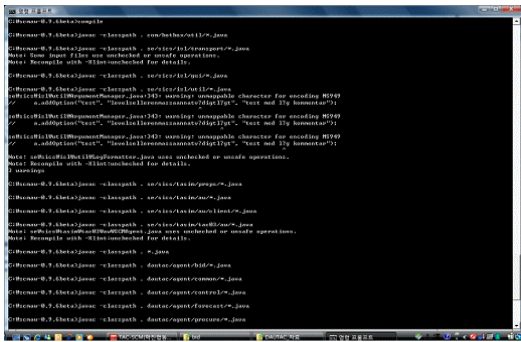


래의 그림 4와 같이 TAC-SCM 홈페이지 (<http://www.sics.se/tac/>)에서 다운받을 수 있다. Java 2 Edition 1.4 이상을 요구하며 이름과 전자메일 주소를 입력한 후 API로 만들어진 “TAC SCM AgentWare”를 다운받을 수 있다. 고안된 알고리즘은 이것을 이용하여 구현하여 게임서버와 메시지를 주고받게 된다. DAUTAC은 전술한 바와 같이 6개의 모듈을 Java를 이용하여 구현하였다.



[그림 4] TAC-SCM 홈페이지

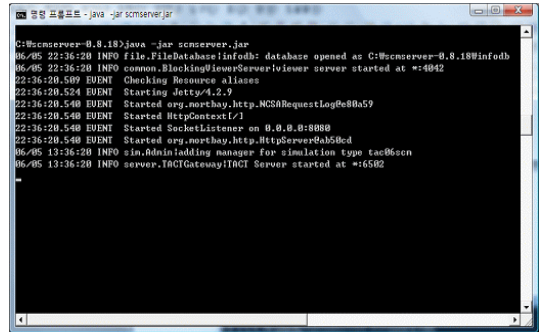
실험을 위해서는 먼저 서버프로그램을 다운받아 설치하여 실행시켜야 한다. 게임서버인 “TAC SCM Server”는 그림 4와 같은 방법으로 다운 받을 수 있다. 자신의 에이전트를 구현하기 위해서는 먼저 다운받은 프로그램을 적당한 폴더에 압축을 푼다. 알고리즘을 코딩한 후에는 다운받은 프로그램과 코딩한 프로그램을 그림 5와 같이 함께 컴파일 한다. 컴파일을 하고 나면 클래스 파일이 생성되고 또한 “scmaw.jar”란 파일이 생성된다.



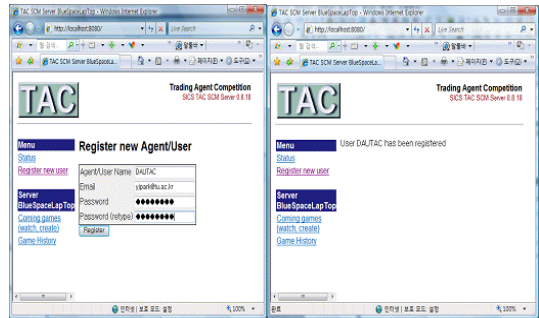
[그림 5] DAUTAC2008을 컴파일 하는 화면

다운받은 압축파일을 적당한 폴더에 압축을 푼다. 다음 그림 6과 같이 명령어 실행창에서 “java -jar

scmserver.jar” 라고 입력하여 실행하면 게임서버가 실행이 된다.



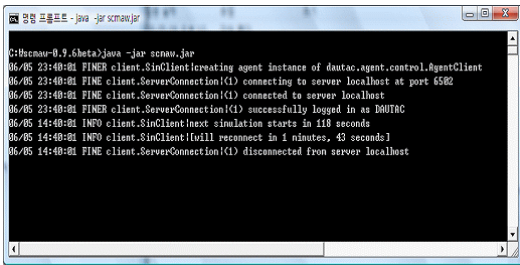
[그림 6] TAC-SCM Server 실행 화면



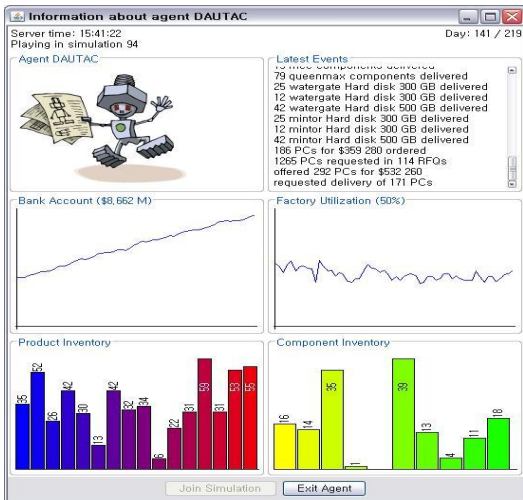
[그림 7] TAC-SCM Server 웹 페이지

서버가 실행되고 있는지를 확인하려면 웹 브라우저를 열고 주소창에 “http://localhost: 8080”라고 입력한 후 확인하면 그림 7과 같이 서버가 실행되고 있는 웹 페이지가 열리면 에이전트 이름, 전자메일주소, 패스워드 등을 입력한 후 등록한다. 다음으로 컴파일된 에이전트를 실행시키면 되는데 에이전트가 있는 폴더에 있는 “aw.conf”파일 에 에이전트명과 패스워드를 지정해 주어야 하는데 이곳의 에이전트 명은 방금 웹 페이지에서 등록한 에이전트 이름과 패스워드로 하면 된다. 이 방법은 로컬에서 실험하는 방법이고 실제 게임은 원격지에서 운영되고 있는 TAC-SCM 공식 서버에 접속하면 된다.

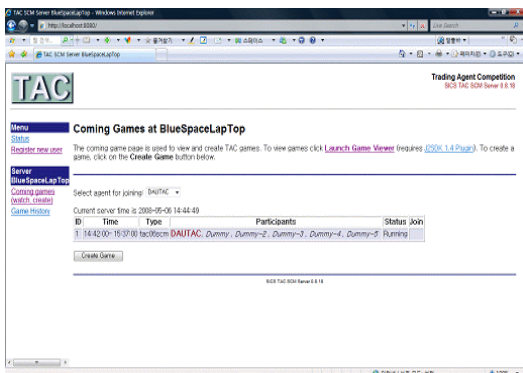
다음으로 그림 8과 같이 명령어 입력창에서 “java -jar scmaw.jar”로 입력한 후 실행시키면 명령어 창에 언제 게임이 시작될 시간에 대한 메시지가 표시됨과 동시에 그림 9와 같이 에이전트가 실행되는 화면이 생성된다. 그림 9에서는 자신의 에이전트 활동만 볼 수 있다.



[그림 8] DAUTAC 실행 명령어 화면

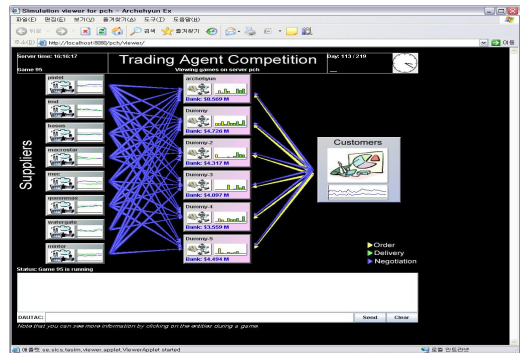


[그림 9] DAUTAC 실행 화면



[그림 10] 진행될 게임에 대한 공지화면

서버 및 다른 에이전트들의 진행상황을 모두 함께 보려면 그림 7에서 "Coming games"을 클릭 한 후 등록된 에이전트 명과 패스워드를 입력하고 그림 10에서와 같이 "Launch Game Viewer"를 클릭하면 그림 11의 전체 진행 상황을 볼 수 있다.



[그림 11] 전체 게임 진행 화면

## 5. 실험 결과 및 결론

### 5.1 로컬 실험

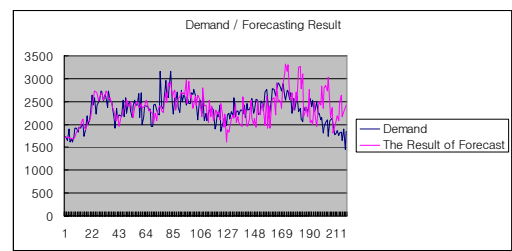
[표 1] 30게임 결과 평균

	DAUTAC	Dummy	
Revenue	14,741,257	58,445,037	
Interest	2,274,652	299,106	
Costs	Material	87,798,012	48,785,320
	Storage	10,768,305	4,324,043
	Penalty	149,538	334,500
Margin 1	14%	10%	
Margin 2	16%	10%	
Result(Net Profit)	18,300,055	5,300,280	

\* Margin 1은 Interest와 Penalty를 제외한 값이며 Margin 2는 Interest와 Penalty를 포함한 값임

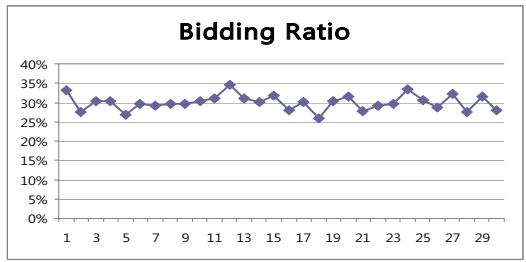
위와 같은 실험을 5개의 dummy agent와 함께 30번을 진행하였다. dummy agent는 서버 측에서 만들어 놓은 것으로 기본적으로 주문제조방식으로 생산하고 있다. 30게임을 진행한 평균 결과를 정리하면 표 1과 같다.

주요 모듈 별로 성능을 분석해 보면 먼저 forecast의 경우 실수율을 잘 예측하고 있다. 30게임 동안의 수요예측을 실수요량의 비율로 표시하면 다음의 그림 12와 같다. 30게임 평균 수요예측비의 값은 약 103%였다.

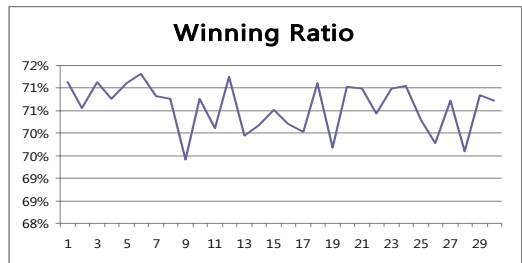


[그림 12] 실수요 대비 수요예측 비율

30게임 동안 bidding 모듈이 고객의 RFQ에 대해 offer 한 비율인 bidding ratio는 그림 13와 같으며 이들 중 고객의 주문으로 이어진 비율 즉 winning ratio는 그림 14와 같다. bidding Ratio와 winning ratio의 평균값은 각각 30%, 71%로 나타났다.

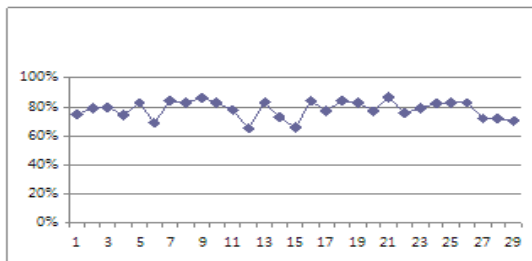


[그림 13] Bidding 모듈의 Bidding Ratio



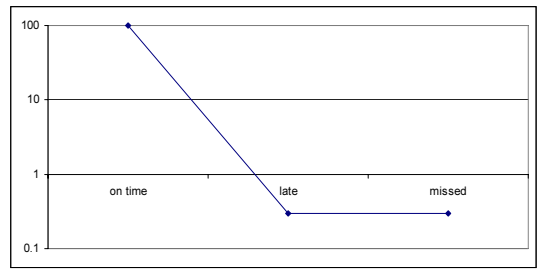
[그림 14] Bidding 모듈의 승률

2,000 cycle/day로 정해진 factory의 30게임 동안의 활용도는 그림 15와 같으며 이의 평균은 78% 정도를 보였다.



[그림 15] Factory Utilization

마지막으로 Delivery를 보면 30게임 평균 정시에 배달한 것이 99.4%, 늦은 것이 0.3%, 납기일보다 5일 이상 지연되어 주문자체가 취소된 것이 0.3%로 나타났으며 이를 그래프로 표현하면 다음의 그림 16과 같다.



[그림 16] Delivery Performance

이상을 종합하면 다음의 표 2와 같다.

[표 2] 30게임 동안의 주요 모듈 평균값

Forecast/Demand	103%	
Winning Ratio	71%	
Bidding Ratio	30%	
Factory Utilization	78%	
Delivery	On Time	99.4%
	Late	0.3%
	Missed	0.3%

표 1 에서 보는 바와 같이 30게임 동안의 순수익을 보면 DAUTAC은 dummy에 비해 3배 정도의 수익을 보였다. 이는 dummy agent가 사용하는 간단한 MTO (Make-To-Order) 방식에 비해서는 본 연구에서 구현한 MTS(Make-To-Stock) 전략이 우수한 것으로 보이며 각 모듈도 잘 작동한 것으로 분석된다. 다만 ATP를 기반으로 고객에게 Offer하기 때문에 배달이 지연되는 경우는 없어야 하는데 실험해서는 지연되는 경우가 0.6% 정도 발생하였다.

## 5.2 2008 TAC SCM Competition

이상의 결과를 바탕으로 2008 TAC-SCM Competition 에 참여하였다. 2008 TAC-SCM에는 그림 17와 같이 17 개의 에이전트들이 참여하였다. qualifying, seeding, quarter final, semi final, final round 순으로 게임이 진행 되었으며 DAUTAC2008은 semi final까지 진출하였으며 semi final B그룹에서 6위를 차지해 3위까지 진출하는 final round에는 참여하지 못했다.



Agent	Leader	Team	Affiliation
1 <b>Botticelli</b>	Amy Greenwald	<b>Botticelli</b>	Brown University, Department of Computer Science
2 <b>Mertacor</b>	Prof. Pericles A. Mitkas	<b>Mertacor</b>	Aristotle University of Thessaloniki
3 <b>BioAgent</b>	Professor Caro Lucas	BioAgent	
4 <b>MinneTAC</b>	John Collins	<b>MinneTAC</b>	University of Minnesota
5 <b>CrocodileAgent</b>	Vedran Podobnik	<b>CrocodileAgent</b>	University of Zagreb
6 <b>TrAder</b>	Bruyere Grégoire	TrAder	IMERIS, Mediteranean Institute of Studies and Research in Computing and Robotics
7 <b>Emo_Sagacious</b>	Professor Caro Lucas	Emo_Sagacious	
8 <b>exam</b>	mhb	exam	
9 <b>TacTex</b>	Peter Stone	<b>TacTex</b>	The University of Texas at Austin
10 <b>DeepMaize</b>	Michael Wellman	DeepMaize	University of Michigan
11 <b>Socrates</b>	Maria Fasli	Socrates	University of Essex
12 <b>GRUNN</b>	Gerben G. Meyer	GRUNN	University of Groningen
13 <b>Merlion</b>	Shih-Fen Cheng	Merlion	Singapore Management University
14 <b>NittanyLISQ</b>	Dr. Soundar Kumara	NittanyLisq	Penn State University
15 <b>NebTAC</b>	Prithviraj (Raj) Dasgupta	NebTAC	University of Nebraska at Omaha
16 <b>DAUTAC2008</b>	Young Jae Park	DAUTAC2008	Tongmyong Univ. & Donga-A Univ.
17 <b>CMieux-2007</b>	Michael Benisch / Norman Sadeh	<b>CMieux-2007</b>	Carnegie Mellon University

[그림 17] 2008 TAC SCM Competition 참여자

[표 3] The Result of Semi Final Group B

Position	Agent	Average Score	Games Played
1	TacTex	8.072	18
2	CMieux	4.912M	18
3	CrocodileAgent	2.490M	18
4	Mertacor	232.299	18
5	NittanyLISQ	-9.491M	18
6	DAUTAC	-33,.24M	18

더미 에이전트와의 실험과는 달리 2008 TAC-SCM Competition에서 DAUTAC은 부품조달에 어려움을 겪었다. 하나의 부품만 모자라도 제품을 생산할 수 없으므로 다른 부품들의 재고는 넘치게 되고 이 결과 재고비용이 높아졌다. 또한 제품제고의 부족으로 bidding할 수 있는 RFQs의 수가 작아져 승률이 낮아졌으며 이러한 악순환으로 인해 적자가 누적되는 결과가 되었다. 현재 DAUTAC의 procurement module을 재분석하여 이러한 문제점을 해결하기 위한 연구를 진행하고 있다.

TAC-SCM 환경에서 고객으로부터 주문을 받기 위해서는 입찰가격에 고객의 수요특성 즉, 수요가 높을 때와 낮을 때, 그리고 재고수준을 고려한 입찰가격의 결정과 입찰량을 능동적으로 결정하는 것이 중요하며, 고객수요의 정확한 예측에 기반한 부품구매 및 재고수준의 결정이 중요한 것으로 생각된다.

## 참고문헌

- [1] Chen, Y.; Peng, Y.; Finin, T.; Labrou, Y.; and Cost, S., "A Negotiation-based multi-agent system for supply chain management," Workshop on Agent-Based Decision Support in Managing the Internet-Enabled Supply-Chain at Agents, 1999.
- [2] Kumar, K., "Technology for supporting supply -chain management," Communications of the ACM Vo.44, No.6, pp.58-61, 2001.
- [3] Sadeh, N.; Hildum, D.; Kjenstad, D.; and Tseng, A., "Mascot: an agent-based architecture for coordinated mixed-initiative supply chain planning and scheduling," Workshop on Agent-Based Decision Support in Managing the Internet-Enabled Supply-Chain, at Agents, 1999.
- [4] David Pardoe and Peter Stone., An Autonomous Agent for Supply Chain Management, 2006.
- [5] John Collins, Raghu Arunachalam, Norman Sade, Joakim Eriksson, Niclas Finne, and Sverker Janson, The Supply Chain Management Game for the 2007 Trading Agent Competition, 2006.
- [6] Mcgee, K., Gartner Research Note DF-22-2973, 2004
- [7] Raskino. M., Gartner Research Note COM- 21-3312, 2004.
- [8] 삼성SDS IT컨설팅실, RTE를 위한 BPM 구현전략. 삼성SDS, 2004.
- [9] 장강일, 실시간 기업(RTE: Real-Time Enterprise), LG 주간경제, p. 15, 2004.

## 박 영 재(Young Jae Park)

[정회원]



- 1999년 2월 : 동아대학교 대학원 경영정보학과 (경영학석사)
- 2004년 2월 : 동아대학교 대학원 경영정보학과 (경영학박사)
- 2006년 3월 ~ 2010년 2월 : 동명대학교 경영정보학과 교수

<관심분야>

경영정보시스템, 전자상거래, 항만물류시스템