

# 생산성 향상의 기본 인프라

지난 호에는 인터넷 유통업의 대표적인 형태인 인터넷 쇼핑물 운영을 위한 내부 프로세스 에이전트를 정의하였다. 또한 인터넷 쇼핑물의 후방 프로세스를 분석해 보았으며 이를 토대로 다양한 운영층 프로세스 에이전트 유형과 특성을 정의하였다. 또한, 다수의 에이전트로 구성된 프로세스 에이전트 팀 조직과 활동 원칙도 제시하였다. 이번 호에는 인터넷 쇼핑물 후방 프로세스 에이전트의 참조 구축 프레임워크를 제시한다.

■ 박광호/한양대학교 경영학부 교수

## 연재 순서

- 1 인터넷 쇼핑물 후방 프로세스 에이전트
- 2 인터넷 쇼핑물 후방 프로세스 에이전트 & 참조 구축 프레임워크 - 이번호

## 프로세스 에이전트 참조 구축 프레임워크

### 구조

인터넷 쇼핑물을 위한 프로세스 에이전트 참조 구축 프레임워크는 <그림 1>과 같이 정의, 운영, 구현 등 3개 영역으로 구성된다. 정의 영역은 이미 전장에서 설명한 바와 같이 프로세스 에이전트에 대한 유형 분류를 제공하고 있다. 운영 영역은 개별 프로세스 에이전트의 활동 사이클과 프로세스 에이전트 팀의 운영 통제 방법을 제공한다.

마지막으로, 구현 영역에서는 정의, 운영 영역에서 제공된 개념적 설계에 대한 구현 방법을 제공하고 있다. 프로세스 에이전트는 기본적으로 객체로 구현되며 COM(Component Object

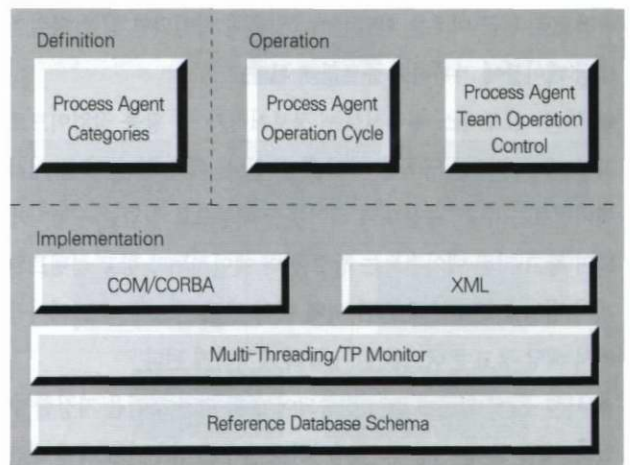
Model) (Box, 1998; Sessions, 1998; Grims et al. 1998)이나 CORBA(Common Object Request Broker Architecture) (Orfali, et al., 1996)와 같은 표준 인터페이스 방식을 채택함을 전제로 한다.

B-to-B에 있어 문서 전달은 XML(eXtensible Markup Language) (Pardi, 1999)을 사용하게 된다. 또한, 운영과 통제를 위한 멀티스레딩, 대용량 트랜잭션 처리를 위한 TP Monitor, 에이전트 간의 대화를 위한 참조 데이터베이스 스키마를 제공한다.

### 개별 프로세스 에이전트 활동

프로세스 에이전트는 웹 페이지 상에서 발생하는 고객과의 인터페이스를 환경 변화로 보고 이에 대해 적절한 대응을 하게 된다. 따라서, 프로세스 에이전트는 인터넷 쇼핑물 프로세스의 해당 프로세스 담당자를 대신하여 활동하게 된다. 본 글에서 제시하는 프로세스 에이전트는 기본적으로 후방 프로세스 담당자에 대한 대리인이며 다음과 같은 특징을 가지고 있다.

- 자율적(Autonomous)이다: 다른 에이전트의 직접적인 통제



<그림 1> 프로세스 에이전트 참조 구축 프레임워크

를 받지 않는다.

- 병렬적(Parallel)이다: 다른 에이전트와 동시에 활동한다.
- 적응적(Adaptive)이다: 모든 에이전트는 자신의 지능으로 적절히 환경 변화에 적응하며 자신도 환경에 영향을 주게 된다.
- 비동기적(Asynchronous): 에이전트는 각자 활동 착수 시점에 개별적으로 활동에 착수하며 다른 에이전트와 작업 보조를 맞출 필요가 없다.
- 순환적(Cyclic): 에이전트는 지속적으로 반복해서 활동한다.

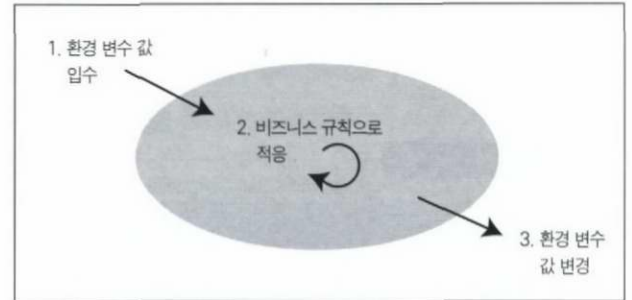
모든 프로세스 에이전트는 기본적으로 대기 상태에 있으며 환경에서 이벤트가 발생하면 이에 따라 활동에 들어 가게 된다. 프로세스 에이전트의 활동은 기본적으로 해당 트랜잭션이 종료될 때까지 지속되며 중간 과정에서 발생하는 모든 활동 결과를 데이터베이스에 기록하게 된다.

예를 들어, 주문 처리 에이전트는 주문 레코드가 신규로 추가되면 활동에 들어가게 된다. 주문 레코드의 신규 추가는 이벤트 발생으로 연결되며 이 때, 해당 트랜잭션을 구별하기 위해 PCI 키를 지정하게 된다. 에이전트는 내부 비즈니스 규칙으로 지속적으로 외부 환경에 대응하게 되는데 비즈니스 규칙을 다음과 같이 정의할 수 있다.

- 활동 시기: 언제 활동해야 하는지 시점을 지정할 수 있다. 예를 들어, 매 1시간마다 작업을 수행한다는 작업 간격을 지정하거나 작업일별로 특정 시간을 지정할 수 있다.
- 완수 조건: 프로세스 에이전트가 일단 활동에 착수하게 되면 작업을 완수할 때까지 지속적으로 활동하게 된다. 따라서, 자신의 활동을 종료할 수 있는 작업 완수 조건이 지정된다. 예를 들어, 입금 에이전트는 무통장으로 입금 예정인 주문건에 대해 지속적으로 입금 여부를 확인한 후, 입금이 확인되면 입금 사실을 해당 테이블에 기록하고 종료하게 된다.
- 대상: 프로세스 에이전트는 정보처리가 주 활동 작업이므로 작업 대상에 대한 규칙을 지정할 수 있다. 예를 들어, 주문 처리 에이전트는 주문, 주문내역 테이블을 대상으로 작업을 수행하게 되며 광고 지원 에이전트는 광고 전략 에이전트가 광고 콘텐츠와 게시 방법을 변경하게 되면 이에 따라 관련 광고 콘텐츠 테이블에서 해당 광고 콘텐츠를 읽어 들여 게시하게 된다.
- 사전 조건: 활동에 앞서 만족되어야 할 사전 조건을 지정할 수 있다. 예를 들어, 신용 승인 에이전트는 주문 접수된 고객에 대한 신용 승인을 처리함에 있어 반드시 이미 회원으로 등록된 고객에

대해서만 가능하다는 사전조건을 정의할 수 있다.

- 작업 규칙: 활동에 대한 작업 규칙을 지정할 수 있다. 예를 들어, 신용 승인 시 신용 한도 초과 여부에 따라 승인을 거부할 수 있으므로 반드시 신용 한도를 넘지 않았는지 확인한다. 또한, 과거에 신용 상태 관련 규칙으로 신용 승인을 결정한다. 또한, 입금 에이전트는 무통장 입금이 확인되면 입금 플래그(Flag)을 설정하여 출고 에이전트가 활동에 들어 가도록 한다.
- 취소 방법: 작업에 대한 적절한 취소 방법을 지정해야 한다. 예를 들어, 주문을 취소해야 한다면 신용 승인 작업, 발주 등 관련 에이전트의 작업을 취소할 수 있어야 한다.
- 예측 지식: 활동 결과에 따른 예측 지식을 지정할 수 있다. 예를 들어, 주문등록 후 2일 이내에 생산계획에 반영되지 않으면 생산 리드 타임은 2주가 넘게 되어 납기를 보장할 수 없게 된다는 예측적 성격의 지식을 지정하는 것이다.



(그림 2) 프로세스 에이전트의 단위 활동 주기

프로세스 에이전트는 일단 활동에 들어 가면 (그림 2)와 같이 환경 변수 값을 입수하고 내부 비즈니스 규칙을 적용하여 환경 변수의 값을 변경하는 일련의 작업을 반복해서 수행하게 된다. 환경 변수는 데이터베이스의 테이블을 지칭하며 환경 변수 값 입수는 지정된 테이블로부터 레코드를 조회하는 것이며 환경 변수 값 변경은 지정된 테이블에 레코드를 추가, 수정, 삭제하는 것이다.

### 프로세스 에이전트팀 활동

프로세스 에이전트는 팀으로 조직되어 활동하게 된다. 예를 들어, 주문처리 프로세스 에이전트팀은 일반 기업의 영업조직과 같이 지역별, 상품별, 고객별로 조직화할 수 있다. 프로세스 에이전트팀의 활동 원칙을 요약하면 다음과 같다.

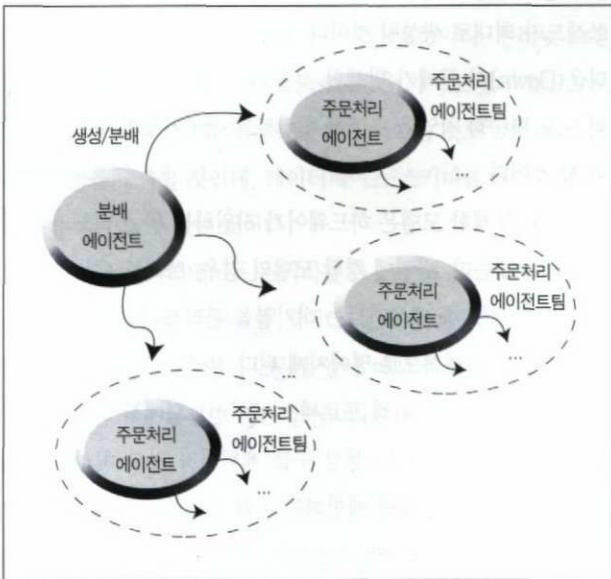
- 분업의 원칙: 프로세스 에이전트는 표준화, 단순화된 프로세스 일부를 수행한다. 이런 분업의 원칙은 일반적으로 Hammer



and Champy(1993)가 지적한 프로세스 조각화(Fragmentation)의 문제점을 안고 있다는 점이 지적될 수 있으나, 에이전트에 의한 프로세스 수행은 지연, 오류의 가능성이 없으므로 무시할 수 있다.

- 프로세스 사슬의 원칙: 프로세스 에이전트는 선행 프로세스 에이전트의 활동 결과에 따라 자신의 담당 업무를 수행한다. 따라서, 각 프로세스 에이전트는 자신의 작업의 완료되면 후행 프로세스 에이전트에게 작업을 인계한다.
- 추적의 원칙: 일련의 프로세스 사슬에 의해 프로세스의 인스턴스가 착수, 완료된다. 우리는 이를 PCI라 지칭하였으며 PCI 키에 따라 특정 프로세스 수행 인스턴스를 추적할 수 있게 된다.
- 무한 생성의 원칙: 프로세스 에이전트팀은 프로세스 인스턴스가 발생하게 되면 생성될 수 있다. 인간 조직과 같이 신규 총원이 필요하지 않는다. 따라서, 피크 시점에 최대의 프로세스 에이전트팀이 생성되어 활동할 수 있다.

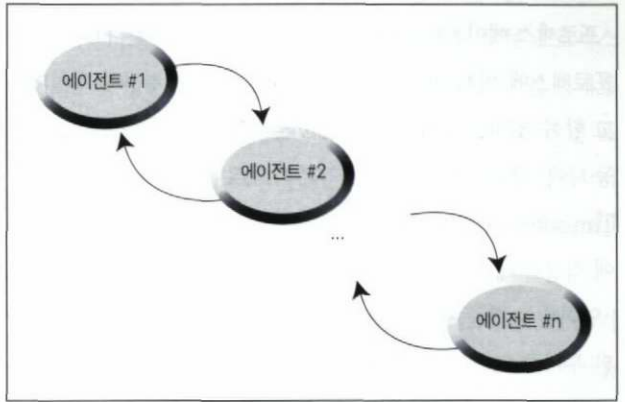
주문처리 프로세스 에이전트팀의 경우, 미처리 되지 않은 주문 건수만큼 에이전트팀이 생성되므로 수백만개의 에이전트가 활동할 수도 있다. 우리는 프로세스 에이전트팀의 통제와 스케줄링을 담당하는 분배 에이전트(Distributor Agent) 개념을 <그림 3>과 같이 도입하였다. 분배 에이전트는 일단 프로세스 인스턴스의 생성을 통보 받게 되면 이를 담당할 프로세스 에이전트팀을 생성하거나, 배분하는 역할을 담당한다. 분배 에이전트는 에이전트팀의 생성 방법에 따라 활동 방법이 결정되는데 다음 장에서 자세히



<그림 3> 프로세스 에이전트팀 활동

논하기로 한다.

분배 에이전트에 의해 프로세스 에이전트 팀이 일련의 프로세스 사슬에 대한 처리를 담당하게 되면 착수 프로세스 처리로부터 시작하여 활동에 들어가 최종 프로세스가 종료되면 다음 작업에 착수하게 된다. 프로세스 에이전트 팀의 내부 활동은 선행 프로세스 에이전트부터 순차적으로 또는 병렬적으로 활동에 들어가게 된다. 만일 후행 프로세스 에이전트의 활동이 취소되면 역으로 선행 프로세스 작업이 진행되기도 한다.

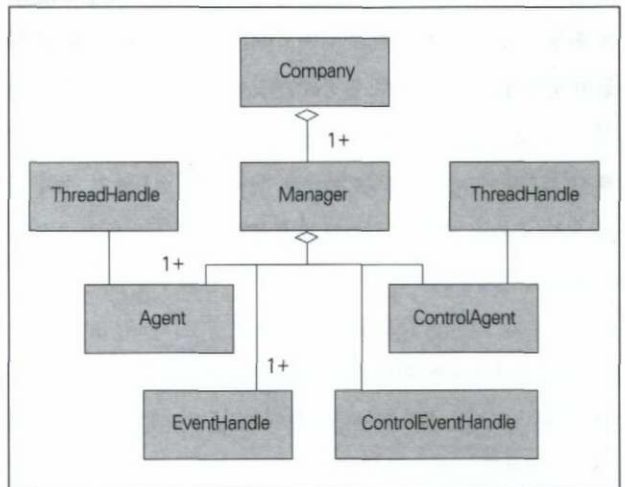


<그림 4> 프로세스 에이전트 팀 내부 활동

## 구현

### 에이전트 기업 클래스 구조

프로세스 에이전트로 구성된 인터넷 쇼핑몰의 내부 구조는 <그림 5>에서 보는 바와 같이 기업(Company), 부서장(Manager), 에이전트, 통제 에이전트(Control Agent)로 구현된다. 기업은 다수의 관리자(관리자란 하나 이상의 프로



<그림 5> 에이전트 기업 클래스도

세스 사슬을 담당하는 에이전트 집합을 통제하는 모듈이다.

따라서, 관리자는 다수의 에이전트, 하나의 컨트롤 에이전트, 다수의 이벤트 핸들(Event Handle), 하나의 컨트롤 이벤트 핸들(Control Event Handle)로 구성된다. 이벤트 핸들은 에이전트에 대한 통제를 위해 설정한 이벤트로 모든 에이전트에 대해 하나씩 생성한다. 각 에이전트는 하나의 스레드 핸들을 갖는다. 이는 에이전트가 스레드(Thread)로 구현되기 때문이다.

### 에이전트 구현 및 활동

프로세스 에이전트는 내부적으로 스레드로 구현된다. 스레드는 프로세스에 비해 상대적으로 오버헤드가 적은 경량의 작업이라고 할 수 있다(Hughes and Hughes, 1997). 다수의 에이전트가 동시에 활동해야 하므로 기본적으로 멀티쓰레딩(Multi-Threading) 기술을 활용한다. 경량의 스레드가 다수의 프로세서에 적절히 배분되어 최적의 처리 속도를 보장하게 되는 것이다.

인터넷 쇼핑물 프로세스별로 부서장과 다수의 에이전트로 구성된 부서를 조직할 수 있음을 밝힌 바 있다. 각 부서별 특성에 따라 각기 다른 컨트롤 방법이 사용될 것이다. 부서별 컨트롤 방법을 구현하기 위해 컨트롤 에이전트를 추가하였다. 컨트롤 에이전트는 전장에서 설명한 분배 에이전트가 구현된 것이다. 일단 부서의 업무가 시작되면 컨트롤 에이전트의 통제에 따라 에이전트들이 각자 주어진 자신의 임무를 수행하게 되는 것이다. 통제의 기본 방법으로 이벤트를 사용하였다.

에이전트팀 활동의 통제를 위해 각 스레드는 이벤트를 발생시키고 이 이벤트의 발생 상태에 따라 활동에 들어가거나 대기하게 된다. 에이전트팀의 활동 모델은 다음과 같이 분류할 수 있다.

- 동기화(Synchronous) 모델: 모든 에이전트들이 같은 기준으로 활동 보조를 맞춘다. 예를 들어, 시간적 기준에 따라 에이전트들이 같은 순서, 사이클로 활동하는 것이다. 이 모델은 시뮬레이션 목적으로 사용되는 통제 모델이다.
- 비동기화(Asynchronous) 모델: 모든 에이전트들이 각자 다른 활동 보조를 맞춘다. 인터넷 쇼핑물의 경우, 비동기화 모델을 사용한다.

하드웨어의 용량에 따라 스레드의 수를 제한할 필요가 있다. 우선, 개별 프로세스 팀 내부의 구성 에이전트의 생성과 소멸 방법을 다음 2가지 모델로 분류하였다.

- 활동 모델(Active Agent Model): PCI 상에서 현재 활동하는

에이전트(Active Agent)만을 생성하고 선, 후행 에이전트는 소멸시키는 모델이다. 이 모델에서는 일단 컨트롤 에이전트가 첫번째 에이전트를 생성하게 되면 순차적으로 후행 에이전트를 생성하게 된다. 만일, 현행 에이전트의 활동이 취소되면 선행 에이전트를 생성할 수도 있다.

- 수면 모델(Sleep Model): 프로세스 에이전트 팀을 구성하고 있는 모든 에이전트가 생성되고 단지, 활동하지 않는 에이전트는 수면에 들어가 대기하는 방법이다.

활동 모델은 스레드 수를 항상 최소로 유지하지만 스레드를 생성, 소멸해야 하는 비용이 발생하게 된다. 수면 모델은 초기에 모든 스레드를 모두 생성하므로 생성, 소멸 비용은 적으나 반면에 스레드 수는 통제되지 않는다. 최대 생성될 스레드 수가 하드웨어 용량에 대비하여 무리가 없을 경우, 수면 모델을 사용하였다.

또한, 프로세스 에이전트팀 생성에 대한 다음 2가지 모델을 분류하였다.

- 무제한(Unlimited) 모델: 새로운 트랜잭션이 발생할 때마다 프로세스 에이전트팀을 생성한다.
- 경합(Bidding) 모델: 제한된 프로세스 에이전트팀이 경합하여 새로운 프로세스 처리에 착수하게 되는 모델로 스레드 숫자를 제한하기 위해 사용된다. 발생된 프로세스 인스턴스는 적절한 데이터 구조 속에 대기하게 된다.

무제한 모델의 경우에 스레드 수의 제한이 없다. 피크 시점에 스레드가 최대 생성될 것이다. 다만, 피크 시점에 있어 시스템 다운(Down)의 문제가 발생할 것을 대비하여 TP 모니터가 적절한 로드 평준화 작업을 수행하여 컨트롤 에이전트의 수를 제한해야 할 것이다.

그러나, 무제한 모델은 하드웨어가 지원하는 한 스레드의 수를 제한하지 않는다. 반면에 경합 모델의 경우, 스레드 수는 제한되며 따라서, 컨트롤 에이전트는 대기열을 관리해야 하며 프로세스 처리 속도는 상대적으로 떨어지게 된다. 따라서, 경합 모델의 이런 단점을 보완하기 위해 프로세스 에이전트팀에서 비교적 처리 기간이 짧은 에이전트는 생성 수를 제한하지 않고 처리 기간이 긴 에이전트는 생성 수를 제한하는 혼합 모델을 사용할 수 있다.

예를 들어, 주문처리 에이전트팀의 경우, 주문 접수, 신용승인, 발주 등은 데이터 확인 절차만으로 처리될 수 있으므로 생성 수



에 제한을 두지 않고 입금, 출고 에이전트와 같이 데이터의 변화(입금, 재고 보충)를 지속적으로 확인해야 하는 에이전트의 경우는 생성 수에 제한을 두는 것이다. 또한, 쓰레드 수 한도를 정해 놓고 한도 내에서는 무제한 모델을 사용하고 한도를 넘어 설 경우, 경합 모델을 사용하는 방법을 사용할 수도 있다.

### 참조 데이터베이스 스키마

후방 프로세스 에이전트는 전방 프로세스에서 입력된 트랜잭션 데이터를 중심으로 작업을 수행하게 된다. 따라서, 후방 프로세스 에이전트는 데이터베이스를 가운데 두고 상호 대화한다고 볼 수 있다. 후방 프로세스 에이전트의 표준화는 결국, 인터넷 쇼핑물의 내부 데이터베이스 스키마의 표준과 연결된다. 이는 B-to-C 비즈니스 모델에서 설명한 다수의 참여 집단의 데이터베이스의 표준과 공개를 요구한다.

그러나, 다수의 참여 집단에 대한 엄격한 데이터베이스 표준화 요구는 무리가 따른다는 지적이 있을 수 있다. 이 문제를 해결하기 위해 마이크로소프트 사이트 서버와 같은 상용 전자 상거래 구축 도구에서는 참조 데이터베이스 스키마를 제공하여 이를 사용하거나 수정하여 사용할 수 있도록 하였다. 또한, 데이터베이스 표준화에 의한 에이전트 간 의사소통 보다는 메시지에 의한 의사소통이 보다 현실적인 해결안이라는 지적도 있다. 이 경우, 메시지 스키마와 메시지 처리를 담당할 컴포넌트의 표준화가 요구된다.

이와 같은 지적에도 불구하고 본 연구에서는 엄격한 데이터베이스 표준화를 전제로 한다. 이 과제는 인터넷 쇼핑물의 정보 인프라 구축과 가상 기업 실현의 관건이기 때문이다. 인터넷 쇼핑물은 전통적인 시스템(Legacy System)을 가지고 있지 않으므로 기존의 데이터베이스와의 통합 문제를 가지는 것도 아니며 신 정보시스템을 구축할 것인데, 데이터베이스 스키마를 굳이 달리 가져야 할 필요가 없는 것이다.

또한, 참조 데이터베이스 스키마는 B-to-B의 문서 전달과 같은 의사소통의 표준화에 기여할 것이다. 참조 데이터베이스 스키마는 B-to-B의 기업간 의사소통에 있어 메시지 표준화에 의한 방법보다는 보다 근본적인 접근 방법이라고 볼 수 있다.

### 결론 및 향후 과제

본 글에서 인터넷 쇼핑물의 후방 프로세스를 지원하기 위한 프

로세스 에이전트 유형을 분류하고 참조 구축 프레임워크를 정의하였다. 디지털 처리를 기반으로 인터넷 온라인 유통업의 주요 성공 요인이 신속한 프로세스 처리와 정보 제공에 있다는 것을 인정한다면 프로세스 에이전트에 의한 업무 처리는 필연적인 것이다. 우리는 프로세스 에이전트에 의한 인터넷 쇼핑물 업무 처리의 1단계로서 운영층 수준의 프로세스 에이전트들에 초점을 두었다. 그러나, 장기적으로 인터넷 쇼핑물 후방 프로세스를 에이전트가 처리하는 비율을 지속적으로 증가시켜 나갈 계획이다.

프로세스 에이전트에 의한 프로세스 수행은 인터넷 쇼핑물의 후방 조직 구조의 변화를 유발시킬 것이다. 보다 많은 프로세스 에이전트의 생성은 수평 조직을 실현할 것이며 가상 기업 형태로 귀결될 것이다. 인터넷 쇼핑물 뿐만 아니라 전자 상거래 후방 프로세스와 조직 구조에 대한 연구는 표준화, 정형화를 발전시켜 나갈 것이며 이것은 프로세스 에이전트의 활용을 확산하는 기반이 될 것이다.

본 논문은 프로세스 에이전트의 구현 방법으로 멀티쓰레딩 기술을 제시하였다. 이는 멀티 프로세서 하드웨어 기술의 발전을 전제로 한 것이다. 멀티 프로세서의 기술 발전은 보다 효과적인 프로세스 에이전트 운영 방법의 향상을 기대할 수 있게 한다.

그러나, 프로세스 에이전트에 의한 인터넷 쇼핑물 운영이 정상 궤도에 오르기 위해선 아직도 해결해야 할 과제가 남아 있다.

첫째, 관련 참여 집단 사이의 인터페이스 표준화 과제이다. 보다 많은 프로세스가 아웃소싱 되는 가상기업의 형태를 띤 인터넷 쇼핑물에 있어 데이터베이스 표준과 인터페이스는 반드시 실현되어야 하는 과제이다.

둘째, 멀티쓰레딩 기술의 지속적인 검증과 효과적인 생성, 소멸 방법의 도출이다. 아직, 상대적으로 활용 경험이 적은 멀티쓰레딩 기술은 운영환경, 하드웨어 환경과 밀접한 관계가 가지고 있다. 따라서, 하드웨어 발전에 따라 효과적인 멀티쓰레딩 운영 방법에 대한 지속적인 연구가 뒷받침되어야 한다.

셋째, 프로세스 에이전트 사이의 의사소통이 데이터베이스를 중심으로 한 간접적인 방법에 의존하고 있는데, 데이터베이스와 직접적인 관련이 없는 의사소통의 경우, 메시지 프로토콜에 기반을 둔 직접 의사소통 방식을 고려해야 할 것이다. 마지막으로, 프로세스 에이전트 유형에 대한 지속적인 추가 정의가 요구된다. 운영층의 프로세스 에이전트팀의 운영을 거쳐 문제점을 보완하고 전략층을 위한 지능형 에이전트를 정의, 구현할 필요가 있다. 