

# 인터넷 마켓플레이스를 위한 XML 기반 계획 에이전트의 설계와 구현

이 용 주<sup>†</sup>

요 약

인터넷 마켓플레이스에서는 사용자를 도와주는 계획 에이전트가 가장 중요한 역할을 수행한다. 최근에 인터넷과 분산처리 기법들의 기술 발전으로 많은 인터넷 마켓플레이스들이 구축되고 있으나 이들 중에서 계획 에이전트에 대한 구체적인 구현 시스템은 아직 없다. 본 논문에서는 인터넷 마켓플레이스를 위하여 XML을 기반으로 한 계획 에이전트를 설계하고 구현한다. 이를 위해 관련 분야인 다중데이터베이스 시스템과 워크플로우 관리시스템의 특징을 비교 분석하고, 계획 에이전트의 기능과 역할을 기술한다. 설계된 계획 에이전트 시스템은 COM+, ASP, XML을 사용하여 구현하며 실제 운영되고 있는 데이터를 얻어 실험을 수행한다.

## Design and Implementation of an XML-based Planning Agent for Internet Marketplaces

Yong-Ju Lee<sup>†</sup>

ABSTRACT

A planning agent supporting customers plays a distinguished role in internet marketplaces. Although several internet marketplaces have been built with the maturity of tools based on internet and distributed technologies, there has been no actual study up to now with respect to the implementation of the planning agent. This paper describes the design and implementation of an XML-based planning agent for internet marketplaces. Since implementing internet marketplaces encounter problems similar to those in other fields such as multidatabase or workflow management systems, we first compare those features. Next we identify functions and roles of the planning agent. The planning agent is implemented using COM+, ASP, and XML, and demonstrated using real data used in an existing system.

**키워드 :** 인터넷 마켓플레이스(Internet Marketplaces), XML, 계획 에이전트(Planning Agent), 서비스 검색 알고리즘(Service Retrieval Algorithm), 인터넷 서비스 임대(Internet Service Leasing)

### 1. 서 론

인터넷 마켓플레이스(Internet Marketplaces)는 최근에 일반 이용자들에게 데이터와 소프트웨어를 쉽게 이용할 수 있도록 지원하는 하나의 새롭고 흥미로운 모델로서 제안되고 있다[1]. 인터넷 마켓플레이스에서 공급자는 인터넷 상에 데이터나 소프트웨어를 제공하고, 수요자는 필요할 때 이러한 데이터나 소프트웨어를 대여한다. 인터넷 마켓플레이스 서비스는 인터넷을 통해 제품을 구매하고 택배와 같은 배달에 의해 제품을 받는 기존의 전자상거래나 온라인 쇼핑과는 달리, 인터넷에 의해 서비스를 요청하고 인터넷으로부터 서비스를 받는다. 인터넷 마켓플레이스는 공급자 입

장에서는 인터넷 상의 수 많은 잠재 고객을 확보할 수 있으며, 수요자 입장에서는 고가의 소프트웨어나 데이터베이스를 구입하지 않아도 되고, 구입 후 유지·관리·보수 등을 걱정할 필요가 없어 그 수요가 급증할 예정이다.

지금까지 하나의 정보시스템을 구축하기 위해서는 데이터베이스, 소프트웨어, 그리고 하드웨어들을 모두 구매하여 단독시스템 내에 설치하여 운영하여야만 했다. 이러한 정보시스템 구축을 위해서는 많은 노력과 투자가 필요하며, 설치된 시스템에 대한 주기적인 유지 및 보수가 요구되고 소프트웨어 운영을 위한 전문적인 훈련이 필요하다. 하지만 이러한 노력과 투자에 비해 사용자는 설치된 소프트웨어의 모든 기능을 다 사용하는 것이 아니라 그들 중 극히 일부분의 기능만 활용하는 경우가 대부분이므로 낭비적인 요소가 많았다. 반면에 인터넷 마켓

<sup>†</sup> 정 회 원 : 성주대학교 컴퓨터공학부 교수  
논문접수 : 2000년 9월 19일, 심사완료 : 2001년 4월 24일

플레이스 사용자는 필요할 때 필요한 기능만 임대할 수 있으며, 상황 변화에 따라 공급자를 자유롭게 변경할 수 있다. 또한 보다 복잡한 문제 해결을 위해 다양한 공급자들로부터 다양한 서비스를 쉽게 제공받을 수 있다. 오늘날 많은 문제들이 여러 기관들로부터 도움을 받아 문제를 해결해야 되는 경우가 종종 있기 때문에 이 기능은 특히 유용하다.

최근엔 인터넷과 분산처리 기법들(예, COM+, CORBA, EJB)의 기술 발전으로 인해 인터넷 마켓플레이스의 구축이 현실화 됨에 따라 많은 시스템들이 활발히 개발되고 있다[2]. 일반적으로 인터넷 마켓플레이스에서는 데이터와 소프트웨어 서비스를 제공한다. 데이터 서비스는 사용자들에게 인터넷 상에서 데이터의 검색 및 그의 활용을 지원하고, 소프트웨어 서비스는 사용자들에게 소프트웨어나 그 소프트웨어를 수행하기 위한 컴퓨터 하드웨어 사용을 지원한다. 인터넷 마켓플레이스에서는 수 많은 수요자와 공급자가 관련되어 있기 때문에 여러 사이트에 분산 저장되어 있는 서비스들을 쉽게 검색할 수 있어야 하고 필요에 따라 이러한 서비스들을 적절히 결합하여야 할 필요가 있다. 이러한 문제 해결을 위해 인터넷 마켓플레이스에서는 사용자를 도와주는 *계획 에이전트*가 가장 중요한 역할을 수행한다[3]. 일반적으로 수요자는 자신이 원하는 것만 기술하고 계획 에이전트는 어떻게 서비스들이 결합되고 실행되는지 기술하고 있는 실행 계획을 세운다. 즉 에이전트는 수요자 질의를 받아서 어떤 사이트에서 데이터를 얻고 어디에 있는 소프트웨어를 수행하는지를 자동적으로 결정한다.

오늘날 대표적인 인터넷 마켓플레이스 시스템으로서 DecisionNet[4], MMM[5], SMART[6] 등이 있다. 이들 중에서 SMART를 제외하고는 아직 계획 에이전트의 개념을 설명하고 있지 않다. 하지만 SMART에서도 계획 에이전트의 중요성은 강조하고 있으나 이에 대한 구체적인 구현 내용은 없다[6]. MMM에서는 XML(eXtensible Markup Language)을 사용하여 데이터와 소프트웨어를 표현하고 활용하고 있다. 반면에 SMART에서는 아직까지 XML은 사용하지 않고 Java를 기반으로 한 시스템을 구축하고 있다. XML[7]은 산업 표준 웹 언어이고 구조화된 문서의 작성을 지원하므로 인터넷 마켓플레이스 구축 언어로써 적합하다. 따라서 본 논문에서는 지금까지의 연구와는 달리 XML을 기반으로 한 계획 에이전트를 새롭게 설계하고 구현하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 인터넷 마켓플레이스를 관련 분야인 다중데이터베이스 시스템 및 워크플로우 관리시스템과 비교 분석하였고, 3장에서는 계획 에이전트의 기능을 설명하고 이를 설계하였다. 4장에서 계획 에이전트를 구현하고 5장에서 실험 분석하였다. 마지막으로 6장에서 결론을 내렸다.

## 2. 관련분야 시스템 비교

인터넷 마켓플레이스의 관련 분야로써 다중데이터베이스 시스템(Multidatabase System)과 워크플로우 관리시스템(Workflow Management System)을 고려할 수 있다. 이들 관련 시스템들과 인터넷 마켓플레이스를 비교함으로써 인터넷 마켓플레이스가 무엇인지, 이들 분야로부터 어떤 기술을 도입할 수 있는지 명확하게 파악할 수 있다.

다중데이터베이스 시스템은 기존에 존재하는 여러 형태의 데이터베이스 시스템들을 아무런 수정을 가하지 않고 상향식으로 자연스럽게 통합한 전역 분산 데이터베이스 시스템이다[8]. 즉 기존의 데이터베이스 시스템이나 응용 프로그램들을 변경없이 계속적으로 사용하면서 통합된 시스템에서의 새로운 사용자에게는 개별적인 기존 데이터베이스 시스템의 특성이나 차이점을 전혀 느끼지 못하고 마치 하나의 시스템인 것처럼 분산된 모든 자료를 사용 가능하게 해준다. 이를 위해 이질적인 지역 데이터베이스들을 논리적으로 연결하기 위한 스키마 통합이 필요하고 각 지역 트랜잭션에 자치성을 보장하는 것이 중요하다.

워크플로우는 비즈니스 프로세스의 자동화를 의미하며, 워크플로우 관리시스템은 소프트웨어를 이용하여 워크플로우의 수행을 정의, 생성, 관리하는 시스템을 의미한다[9]. 워크플로우는 비즈니스의 목적으로 업무에 참여하는 구성원들 사이에서 이루어지는 문서와 정보 또는 작업의 절차를 정의된 규칙(rule)에 따라 자동화하며, 각 구성원들의 역할(role)과 전체 업무흐름(routing)을 정의하여 워크플로우를 수행하는데 활용된다. 대부분 워크플로우 관리시스템은 비슷한 구조를 가지고 있는데 일반적으로 프로세스 디자이너, 워크플로우 엔진, 그리고 워크플로우 클라이언트로 구성되어 있다[10].

다중데이터베이스 시스템에 비해 워크플로우 관리시스템은 트랜잭션 관리에 대한 문제가 덜 심각하고 질의 최적화가 없다. 워크플로우 관리시스템에서는 관리자가 워크플로우를 정의하고 관리하며 사용자가 자동화 도구를 사용하여 업무를 수행하는데 반해 다중데이터베이스 시스템에서는 사용자가 질의를 요청하면 사용자 간섭없이 질의 최적화가 자동적으로 작업을 수행한다. 즉 워크플로우 관리시스템은 다중데이터베이스 시스템보다 더욱 인간 중심적이라 할 수 있다.

워크플로우 관리시스템은 인터넷 마켓플레이스 구현을 위해 다중데이터베이스 시스템에서는 지원되지 않는 여러 가지 흥미롭고 유익한 기능들을 제공한다. 예를 들면, 데이터베이스 액세스 기능 외에 인터넷 마켓플레이스에서 꼭 필요한 프로세스(소프트웨어) 실행을 지원한다. 그리고 워크플로우 관리시스템에서의 유용한 다른 한가지 개념은 역할의 개념이다. 즉 어떤 서비스들(워크플로우 내의 사람들)

은 프로세스 내에서 어떤 일(task)을 만족하지만 어떤 서비스들은 이 역할을 만족하지 않을 수 있다. 만일 만족하지 않는다면 이 요청을 다른 사람에게 인계하거나 그것을 취소할 수 있다. 인터넷 마켓플레이스에서 이것은 서비스 대행자와 관련되고 동적 환경에서는 특히 중요한 기능이다. 따라서 워크플로우 관리시스템은 다중데이터베이스 시스템에 비해 동적 환경에서 더욱 유연하다고 할 수 있다. 그렇지만 인터넷 마켓플레이스와 비교하면 워크플로우 관리시스템은 자원 검색에 대한 기능이 없다.

요약하면, 워크플로우 관리시스템과 다중데이터베이스 시스템의 많은 아이디어가 인터넷 마켓플레이스에 적용 가능하지만 많은 부분이 수정되어야만 한다. 다중데이터베이스 시스템은 기존에 이미 존재하고 있는 데이터베이스들의 통합을 목표로 하고 있는 반면에 인터넷 마켓플레이스는 가상의 인터넷 서비스들의 활용에 그 목적이 있다. 인터넷 마켓플레이스에서는 인터넷 기반의 동적인 환경 때문에 하나의 통일된 뷰를 제공하기 위한 스키마 통합이 어렵다. 그렇지만 다중데이터베이스의 스키마 통합 기능을 확장시켜 이 기능을 제공한다면 사용자는 수작업으로 서비스들을 통합할 필요없이 이들을 자동화할 수 있을 것이다. 인터넷 마켓플레이스에서는 서비스들을 활용하기 위해 실행 계획이 만들어지며 이들이 저장되어 반복적으로 수행될 수 있다. 이를 위해 다중데이터베이스의 질의 최적화기가 응용될 수 있다. 지금까지 기술한 내용을 요약하면 <표 1>과 같다.

<표 1> 관련분야 시스템 비교

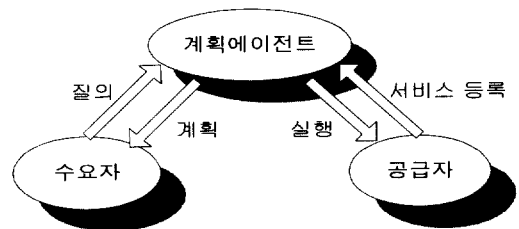
특징	MDBS	Workflow	Marketplace
트랜잭션 관리	복잡	간단	보통
질의 최적화기	있다	없다	계획기로 수정
작업 처리	알고리즘 중심	인간 중심	중간
프로세스 실행	없다	있다	있다
역할 개념	없다	있다	있다
동적 환경	기능 미비	유연	매우 유연
자원 검색	없다	없다	있다
스키마 통합	있다	없다	어렵다
실행 계획	있다	없다	있다

### 3. 계획 에이전트

#### 3.1 계획 에이전트의 기능

인터넷 마켓플레이스의 여러 기능들 중에서 데이터와 소프트웨어 서비스들을 결합하고 실행하기 위해 수행순서(계획)를 만들어서 사용자가 사용하기 쉽도록 지원해 주는 계획 에이전트가 가장 중요한 역할을 수행한다. 계획 에이전트는 수요자에게 서비스들을 즉시 실행할 수 있는 프로그램(계획)을 제공하므로 검색기와는 그 기능이 다르다. 검색기는 어떤 요구사항을 만족하는 서비스의 이름과 간단한 설명만 제공한다. 따라서 계획 에이전트는 다중데이터베이스 시스템에서의 질의 최적화기와 비슷한 역할을 수행한다.

실행 계획을 생성하기 위해서는 메타데이터(metadata)가 이용되는데 이 정보는 공급자가 서비스를 등록할 때 만들어진다. (그림 1)은 계획 에이전트가 수요자 및 공급자와의 전반적인 관계를 보이고 있다.



(그림 1) 계획 에이전트와 수요자/공급자 간의 관계

계획은 포함하고 있는 서비스의 개수에 따라 간단한 계획과 복잡한 계획으로 분류할 수 있다. 간단한 계획은 단지 한 개의 서비스만 호출한다. 반면에 복잡한 계획은 서비스들의 연속적인 호출을 요구한다. 복잡한 계획의 성능은 어떻게 수많은 서비스들을 효율적으로 결합할 수 있는가에 달려있다. 예를 들면, 한 고객은 표준 데이터 형식(예, SDTS 형식)으로 작성된 서울 지역의 전자 지도를 요구할 수 있다. 그렇지만 이 지도를 제공할 수 있는 공급자는 현재 지도 작성을 위해 가장 많이 사용하고 있는 ArcInfo 시스템용 지도만 제공할 수 있다고 가정한다. 그러면 계획 에이전트는 ArcInfo 데이터를 SDTS 데이터로 변환할 수 있는 소프트웨어를 검색하여 두 개의 서비스로 구성된 하나의 계획을 만들 수 있다. 첫번째는 ArcInfo용 데이터를 호출하고 두번째는 데이터 변환 소프트웨어를 호출하는 계획을 만든다.

인터넷 마켓플레이스에서 실행 계획의 작성은 인터넷 기반의 동적인 환경 때문에 기존의 다중데이터베이스 시스템보다 훨씬 복잡하고 구현하기 어렵다. 기존의 시스템들처럼 메타데이터를 사전에 만들기 어렵기 때문에 계획 에이전트는 공급자가 제공하는 등록 정보에 거의 의존하게 된다. 즉 계획 에이전트는 미리 저장되어 있는 정보와 동적으로 수집되는 메타 정보를 이용하여 자동적으로 실행 계획을 생성한다. 메타데이터는 주어진 요구를 만족시키기 위해서 어디에서 작업이 수행되어야 하는지, 어느 알고리즘을 사용해야 하는지, 그리고 작업을 처리하기 위해 어떠한 절차를 따라야 하는지 등을 포함하고 있어야 한다. 작업 절차에 관한 규칙은 시스템 내에서 자동으로 정의할 수도 있겠지만 워크플로우 관리시스템과 비슷하게 관리자에 의해 수동으로 준비될 수도 있다. 한편, 한번 만들어진 계획은 똑같은 작업에 대하여 반복적으로 적용시키기 위해서 고객에 의하여 저장될 수 있도록 한다.

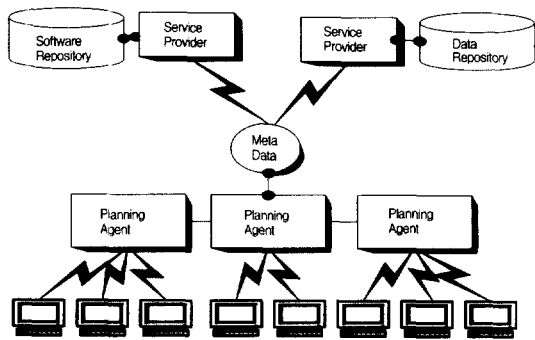
계획 에이전트의 기능이 다중데이터베이스의 질의 최적화기 기능과 비슷하게 보이지만 인터넷에 의해 야기되는 환경 요소 때문에 차이점이 많다. 첫째로 인터넷 마켓플레이스에서는 데이터베이스의 구조에 대한 정확한 지식이 없어도 정

보를 검색할 수 있도록 융통성 있는 질의어를 제공해야만 한다. 예를 들면 유사성(similarity) 검색이나 동등한 서비스(equivalent service) 검색 등을 지원해야 한다. 둘째로 동적 인터넷 특성 때문에 동적인 스키마 통합을 요구한다. 인터넷 상에서는 새로 생기거나, 사라지거나, 인터페이스가 변하는 사이트가 일반적이기 때문에 다중데이터베이스와 같은 단일의 전역 스키마를 구성하기가 매우 어렵다. 셋째로 복잡한 비용 계산이 요구된다. 질의 최적화 단계에서 고려되는 물리적 비용(수행 시간 또는 데이터 처리량) 뿐만 아니라 금전상의 비용(임대비)이나 서비스의 품질도 고려되어야 한다. 그 외 사용량, 수정일, 정확성, 자료 다운로드 성능 등도 고려해야 한다. 즉 다차원 비용 함수의 최적화를 요구한다.

계획 에이전트를 성공적으로 구현하기 위해서는 이질적 분산 환경을 극복해야 한다. 이를 위해 마켓플레이스 서비스들은 산업 표준 방법으로 표현되어야 한다. 인터넷 마켓플레이스를 매력적으로 만들기 위해서는 다양한 서비스를 제공하는 많은 수의 공급자를 확보하는 것이 필수적이지만 수요자들에게 어느 정도 일치된 인터페이스를 제공하는 것이 더욱 필요하다. 분명히 표준화는 이질적 마켓플레이스의 문제점들을 완화 시킬 수 있을 것이다. 특히 인터넷 마켓플레이스 상에서 서비스들은 동적으로 증가되므로 이 시스템은 개방성과 확장성이 고려되어야 한다.

3.2 계획 에이전트의 설계

인터넷 마켓플레이스에서는 많고 다양한 사용자들을 지원하기 위해 이질적 분산 환경 및 개방성과 확장성을 고려해야 한다. 이를 위해 본 연구에서는 XML 문서를 활용한 다층(multi-tier) 구조 분산 시스템을 설계하였다. XML을 이용한 이유는 공급되는 서비스들에 대한 메타데이터로서 사용이 가능하며, 시스템 간에 플랫폼 독립적인 데이터 교환 및 처리가 가능하기 때문이다. 또한 XML은 웹을 이용한 트랜잭션 처리의 확장성을 보장하면서 현존하는 웹 환경 및 도구를 최소한의 변경으로 이용할 수 있는 이점을 제공하기 때문이다. 전체적인 시스템 구조는 (그림 2)와 같으며 이에 대한 설명은 다음과 같다.



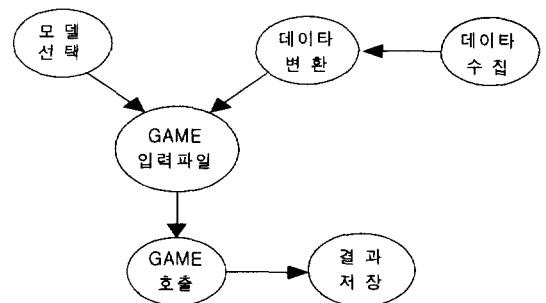
(그림 2) 인터넷 마켓플레이스의 구조

- 서비스 제공자 : 제공되는 서비스 정보를 저장하고 있다. 사용자들에게 서비스 질의 및 실행을 지원한다.
- 계획 에이전트 : 인터넷 마켓플레이스의 미들웨어 (middleware) 역할을 수행한다. 사용자의 요청을 받아들여서 그 요청을 해결하는데 필요한 정보를 검색·조작·통합한다.
- 클라이언트 : 웹 브라우저에 의해 계획 에이전트에 접속한다.

계획 에이전트는 사용자가 증가함에 따라 가변적으로 늘어날 수 있다. 각 구성요소는 COM+(또는 CORBA)와 같은 분산객체 기술을 사용함으로써 플랫폼 독립성을 보장한다. 다른 계층 사이의 정보 교환은 XML 문서 형태로 전송되므로 어떠한 개발 환경이든 상관없이 이질적 분산 서비스가 지원된다.

인터넷 마켓플레이스를 구축하기 위해서는 일반적으로 하나 이상의 특정 응용분야를 선택해야 하는데 본 논문에서는 가장 효과적인 경영과학 도구들 중 하나이고 다른 분야에 비해 비교적 정형화가 잘되어 있는 OR(Operations Research) 분야에 초점을 맞추었다. 그렇지만 본 논문에서의 접근방법은 다른 응용분야에도 쉽게 적용시킬 수 있을 것이다. 부록은 OR 문제들 중 하나인 수송모형을 GAMS(General Algebraic Modeling System)[11]로 표현한 것이다.

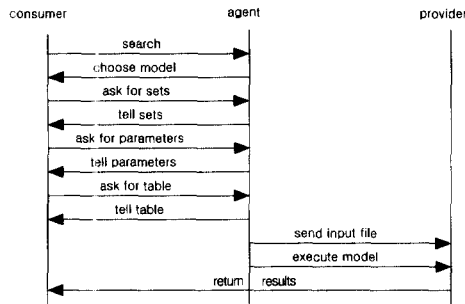
GAMS에 대한 실행 계획은 (그림 3)과 같다. 먼저 수요자는 계획 에이전트로부터 원하는 모델을 검색하여 선택한다. 에이전트는 선택된 모델을 실행시키기 위한 데이터를 수집한다. 이를 위해 에이전트는 메타데이터를 이용하여 데이터 수집 순서를 결정한다. 예를 들면, 수송모델에서 사용자가 집합(sets)을 정하면 이에 대한 파라메타(parameters)와 테이블(table) 입력 폼이 웹 상에 동적으로 생성된다. 입력된 데이터는 GAMS 데이터 타입에 맞게 변환되고, 변환된 데이터를 결합하여 GAMS 입력 파일을 만든다. 에이전트는 GAMS 서버에 입력 파일을 전송하고 GAMS 모델을 실행시킨다. 최종적으로 결과 파일의 URL이 사용자에게 리턴된다.



(그림 3) GAMS 실행 계획

이러한 수행 절차를 구체적으로 설계하기 위해 수요자, 계획 에이전트, 그리고 공급자간 상호 작용을 순차 다이어

그림(sequence diagram)으로 표현하면 (그림 4)와 같다.



(그림 4) 계획 에이전트를 위한 순차 다이어그램

계획 에이전트의 역할이 성공적으로 수행되기 위해서는 효율적인 메타데이터의 구축이 절대적으로 필요하다. 공급자들이 서비스를 등록할 때 그들은 쉽게 서비스를 등록할 수 있고, 쉽게 분류·결합할 수 있는 간단한 인터페이스를 기대한다. 따라서 등록 과정은 웹 브라우저의 폼 형태로 입력되며 공급자는 미리 준비된 분류 항목을 조사하여 등록 내용이 속할 카테고리를 선택한다. 이때 계획 수립을 위한 관련 프로세스 규칙이나 절차 등도 같이 입력한다. 서비스 제공자는 지역적으로 분산되어 있으므로 공급자는 자료를 전송하기 전에 먼저 계획 에이전트에게 자신의 컴퓨터 위치(URL)를 알려준다. 공급자와 에이전트 간에는 XML과 HTTP 프로토콜을 이용하여 통신이 이루어진다. 이러한 표준화된 접근 방법은 각 계층의 컴퓨터가 어떤 소프트웨어, 운영체제, 그리고 어떤 데이터베이스 시스템을 사용하던 간에 상관없이 작동될 수 있다. 마지막으로 계획 에이전트는 공급자로부터 온 XML을 파서(parse)하여 SQL(Structured Query Language) 입력문을 생성하고 이를 실행하여 메타데이터를 구축한다.

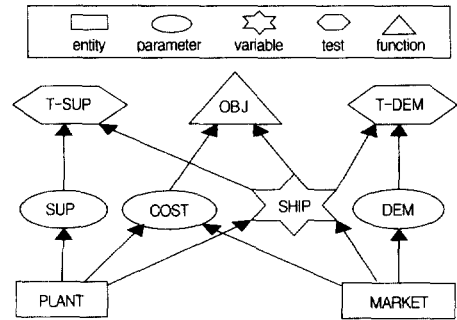
3.3 서비스 검색 알고리즘

본 검색 알고리즘은 OR 문제들의 정형화된 수학적 프레임워크를 제공하기 위해 개발된 SSM(Structured Service Models)[12]에 그 기반을 두고 있다. SSM이란 서비스 정형화를 위하여 사용되는 하나의 특별한 태스크 그래프(directed acyclic graph)로 정의될 수 있다. SSM에서 모든 노드(node)는 항목을 나타내고 모든 아크(arc)는 항목들 간 종속성을 표현한다. SSM은 5개의 노드 타입 (즉, entity, parameter, variable, test, function)으로 구분되며, 아크는 단지 다음과 같은 노드 사이에서만 존재한다

- (1) entity로부터 parameter와 variable 사이,
- (2) parameter 또는 variable로부터 test, function 사이.

따라서 SSM은 3층 노드 (즉, entity 층, parameter와 variable 층, 그리고 test, function 층) 태스크 그래프로 구성되

어진다. 한 예로 부록에 나와 있는 수송모델을 SSM으로 표현하면 (그림 5)와 같이 된다.



(그림 5) 수송모델 SSM

(그림 5)는 entity PLANT와 MARKET로 표현된 공장들(plants)과 시장들(markets) 사이의 운송 관계를 모델링한 것이다. 각 공장의 공급량(supply)과 각 시장의 수요량(demand)은 미리 주어진다. 이는 각각 parameter SUP와 DEM으로 표현된다. 공장과 시장 사이에서 거리 단위당 수송비용은 parameter COST로 표현된다. 이 모델의 결과는 각 공장에서 각 시장으로 운송되는 최적 출하량(shipment), 즉 variable SHIP을 계산하는 것이다. 제약조건으로써 출하량은 공장 생산량을 초과할 수 없고, 시장 소비량을 만족시킬 수 있는 양이어야 한다. 이는 각각 test T-SUP와 T-DEM으로 표현된다. 마지막으로 이 모델의 목적함수는 비용을 최소화시키는 것이다. 이는 function OBJ로 표현된다.

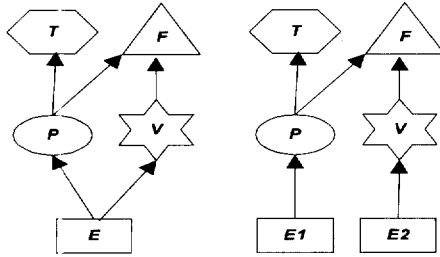
본 논문에서는 서비스 검색 알고리즘을 위해 유사 구조(similarity structure) 일치 메커니즘을 제안한다. SSM 그래프에서 노드 타입의 개수는 1층에 1개, 2-3층에 각각 2개씩 구성되어 있으므로 3층 구조 SSM 그래프 G는 다음과 같이 구조화 될 수 있다.

$$G = (U_1, U_2, U_3, A)$$

여기서,  $U_1 = U_E$ ,  $U_2 = U_P \cup U_V$ ,  $U_3 = U_T \cup U_F$ 로 표현되고,  $U_E$ 는 entity 노드,  $U_P$ 는 parameter 노드,  $U_V$ 는 variable 노드,  $U_T$ 는 test 노드,  $U_F$ 는 function 노드 타입, 그리고 A는 아크를 나타낸다.

SSM 그래프에서 종속관계에 있는 노드  $v, w$  사이에는 항상 아크  $(v, w)$ 가 존재하며, 이들 아크의 종류는 6가지(즉,  $U_E \rightarrow U_P$ ,  $U_E \rightarrow U_V$ ,  $U_P \rightarrow U_T$ ,  $U_P \rightarrow U_F$ ,  $U_V \rightarrow U_T$ ,  $U_V \rightarrow U_F$ )로 분류할 수 있다. 각 분류별 아크의 개수를 계산하면  $i$  번째 반복 개수를 나타내는 6차원 벡터로 표현될 수 있다. 따라서 두개의 SSM 그래프 G와 G'가 주어졌을 때, 분류별 아크의 개수가 일치하면(즉, 벡터  $a, a'$ 가  $\sum_{i=1}^6 (a_i - a'_i) = 0$ 가 되면) 두 그래프는 일반적인 유사성 특성(generic similarity property)[13]을 만족한다. 예를 들어 (그림 6)은 서로 다른 SSM이지만 아크는  $U_E \rightarrow U_P$ ,  $U_E \rightarrow U_V$ ,  $U_P \rightarrow U_T$ ,  $U_P \rightarrow$

$U_F, U_V \rightarrow U_F$  각각 1개씩 존재하므로 분류별 아크의 개수는 일치한다. 따라서 이 두 그래프는 유사 구조라 할 수 있다.



(그림 6) 유사 구조 SSM

유사 구조 일치 메카니즘을 쉽게 적용하기 위해 본 논문에서는 먼저 SSM 그래프를 문자열로 변환한다. 예를 들면,  $U_E$ 와  $U_P$  간 아크를 A,  $U_E$ 와  $U_V$  간 아크를 B,  $U_P$ 와  $U_T$ ,  $U_F$  간 아크를 각각 C, D, 그리고  $U_V$ 와  $U_T$ ,  $U_F$  간 아크를 각각 E, F로 치환하면 (그림 5) 수송모델 SSM은 문자열 "AABABACDEFEC"로 변환될 수 있다. 이를 알파벳 순서로 정렬하면 "AAAABBCCDEEF"가 되어 각 종류별 아크의 개수를 쉽게 파악할 수 있다. 이러한 정렬된 문자열은 공급자가 서비스를 등록할 때, 또는 사용자가 원하는 서비스를 필터링할 때 사용된다. SSM 작성은 비교적 단순하기 때문에 직관적인 그래픽 사용자 인터페이스를 제공할 수 있으며[14], 이러한 도구로부터 자동적으로 문자열을 변환할 수 있다. 하지만 본 논문에서는 자동 변환에 대한 연구는 고려하지 않는다.

#### 4. 구 현

본 연구에서는 계획 에이전트를 구현하기 위하여 마이크로소프트사의 COM+를 사용하였다. COM+ 컴포넌트 서비스는 미들티어(middle-tier) 트랜잭션 관리 메카니즘 및 객체 브로커(broker) 역할을 수행한다. 클라이언트/서버 프로그래밍을 위해 ASP(Active Server Pages)를 사용하였고, ASP와 컴포넌트 사이의 자료 교환을 위해 XML을 사용하였다.

인터넷 마켓플레이스를 구현하기 위해 독일 Humboldt 대학의 MMM(<http://macke.wiwi.hu-berlin.de/mmm/>)으로부터 메타데이터를 얻어, 실제 운영되고 있는 데이터에 기초하여 시스템을 설계하였다. 이 메타데이터에는 두 종류의 메타 객체(즉, DSO와 MSO)가 있는데[15], DSO(Data Service Objects)는 데이터 서비스를 표현하고 MSO(Method Session Objects)는 소프트웨어 서비스를 표현하고 기술한다. DSO와 MSO는 XML 문서로 작성되어 사용자가 서비스를 검색하거나 실행할 때 사용된다. 그렇지만 이들 메타 객체는 계획 수립을 위해 필요한 프로세스 규칙이나 절차 등이 정의되어 있지 않기 때문에 본 연구에서는 이를 확장하여 적용시켰다.

인터넷 마켓플레이스에서 공급자는 메타데이터를 사용하여 서비스를 등록할 수 있어야 하고 수요자는 이 메타데이터를 이용하여 자원을 검색할 수 있어야 한다. 먼저 공급자가 서비스를 등록하고자 할 때 폼 입력력 화면의 HTML 페이지가 클라이언트로 다운로드 된다. 클라이언트에서는 컴포넌트를 사용하여 요구되는 서버 프로그램의 위치를 찾아 XML 파일을 전송한다. (그림 7)은 전송된 수송모델 XML 문서의 예를 보여주고 있다. 서버 프로그램은 클라이언트로부터 오는 XML을 파서하여 SQL 문장을 생성하고, 이를 실행시켜 데이터베이스 입력 작업을 수행한다. 현재 SQL문 실행을 위해 관계형 DBMS를 사용한다. 본 구현에서는 XML을 조작하고 파싱하기 위하여 마이크로소프트사의 MSXML DOM(Document Object Model) 인터페이스를 사용하고 XML의 플랫폼 독립적인 전송을 지원하기 위하여 MSXML IXMLHTTPRequest 인터페이스를 사용한다[16]. 그리고 데이터베이스의 검색, 삽입, 수정을 위하여 ADO (ActiveX Data Object)를 사용한다.

```

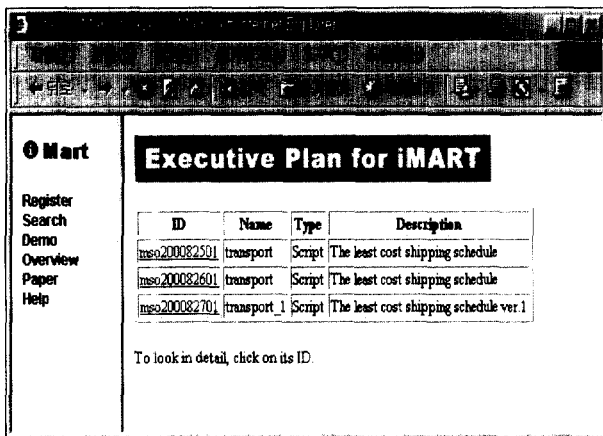
<mso name = "transport" id = "mso200082601">
<general>
  <person type = "provider" lastname = "Rosenthal"
    firstname = "Richard" middle = " ">
    <address> ... </address>
    <email>richard@gams.netlib.org</email>
  </person>
  <creation year = "2000" month = "1" day = "12"></creation>
  <version>1.0</version>
</general>
<access src = "http://www.netlib.org/gams/transport"
  type = "Script" language = "gams"></access>
<interface>
  <call>H03ABE</call>
  <input>
    <set name = "i"><type><array rows = "n"><type>string
      </type></array></type>
    <description>used to declare and define a set
      </description></set>
    <set name = "j"><type><array rows = "m"><type>string
      </type></array></type></set>
    <parameter name = "a"><type><array rows = "n"><type>
      integer</type></array></type>
    <description>a constant or group of constants
      </description></parameter>
    <parameter name = "b"><type><array rows = "m"><type>
      integer</type></array></type></parameter>
    <table name = "d"><type><matrix rows = "n" cols = "m">
      <type>real</type></matrix></type>
    <description>one of the ways of initializing
      parameters</description></table>
  </input>
  <output>
    <variable name = "x"><type><matrix rows = "n"
      cols = "m"><type>real</type></matrix></type>
    <description>shipment quantities
      </description></variable>
  </output>
  </interface>
</mso>
    
```

```

</output>
<ssm>AAAABBCDEEF</ssm>
</interface>
<domain>
  <category path = "/Optimization/Constrained/
  Transportation and Assignment Problems">
  </category>
  <description>The least cost shipping schedule
  </description>
  <keyword>transportation problem</keyword>
  <reference>Dantzig, Linear Programming and Extension,
  New Jersey, 1963</reference>
</domain>
</mso>
    
```

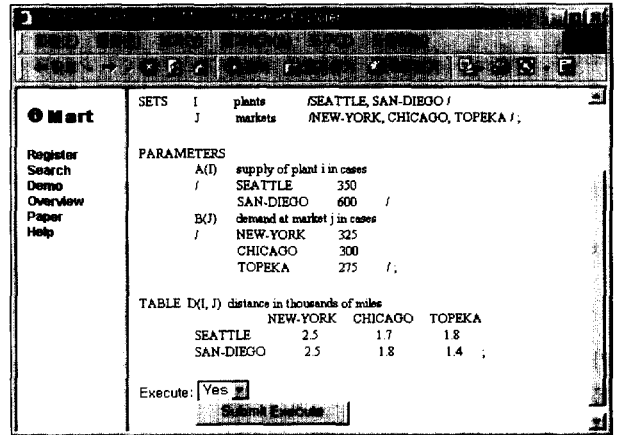
(그림 7) 수송모델 XML 문서의 예

수요자가 웹 브라우저에서 질의를 요청하면 계획 에이전트는 메타데이터 내의 프로세스 규칙을 이용하여 실행 계획을 생성한다. 구체적으로 설명하면 사용자는 찾고자 하는 문제에 대한 SSM을 작성하고 이를 문자열로 변환한 후 계획 에이전트에게 질의를 요청한다. 계획 에이전트는 문자열 검색 알고리즘을 사용하여 패턴이 일치하는 서비스들을 나열한다. 이때 XML로 만들어진 리스트는 XSL(eXtensible Stylesheet Language) 스타일시트를 사용하여 HTML로 변환되어 웹 브라우저에게 보낸다. (그림 8)은 이 결과를 보여주고 있다.



(그림 8) 실행 계획

리스트에 있는 하나의 아이템을 선택하면 계획 에이전트는 메타데이터를 이용하여 입력 데이터를 수집한다. 예를 들어, 집합에서 줄 2개, 열 3개를 입력하면 테이블 입력시표 2x3 HTML 폼이 자동적으로 생성된다. 입력 완료 후 에이전트는 메타데이터 정보를 이용하여 데이터 타입을 변환(예, integer 또는 real로 변환)하고 데이터를 결합하여 입력용 XML 파일을 생성한다. 이것도 XSL 언어를 사용하여 HTML 파일로 변환된 후 브라우저에 디스플레이 된다. 이 결과는 (그림 9)에 표현하고 있다.



(그림 9) GAMS 입력파일

사용자는 자세한 사항을 검토한 후에 이 내용의 실행을 요청한다. 이때 공급자 플랫폼에서 사용자 요청사항에 대한 작업을 실행하고, 적당한 메시지가 포함된 XML 파일을 생성한다. ASP 스크립트는 XSL 스타일시트를 사용하여 XML을 다시 HTML 파일로 변환한 후 그 결과를 사용자에게 보낸다. 이러한 프로그램 처리 과정은 (그림 10)와 같으며 각 단계별로 클라이언트가 관련 메소드들을 계획 에이전트로부터 호출한다.

[1단계] 서비스 질의
<ul style="list-style-type: none"> <li>● 계획 에이전트에게 질의를 요청한다.</li> <li>● 패턴 일치되는 서비스 리스트를 얻는다.</li> <li>● XSL을 사용하여 XML을 HTML로 변환한다.</li> <li>● HTML 파일을 브라우저에게 보낸다.</li> </ul>
[2단계] 입력파일 생성
<ul style="list-style-type: none"> <li>● 리스트 중 하나의 항목을 선택한다.</li> <li>● 입력 데이터를 수집한다.</li> <li>● 입력용 XML 파일을 생성한다.</li> <li>● XSL을 사용하여 XML을 HTML로 변환한다.</li> <li>● HTML 파일을 브라우저에게 보낸다.</li> </ul>
[3단계] 서비스 실행
<ul style="list-style-type: none"> <li>● 실행을 요청한다.</li> <li>● ASP 스크립트는 요청 내용을 기반으로 XML을 작성한다.</li> <li>● 공급자 플랫폼에 그 XML을 보낸다.</li> <li>● 실행한다.</li> <li>● 공급자로부터 실행 결과를 얻는다.</li> <li>● XSL을 사용하여 XML을 HTML로 변환한다.</li> <li>● HTML 파일을 브라우저에게 보낸다.</li> </ul>

(그림 10) 프로그램 처리 과정

(그림 10)에서 중요한 구현 기법은 XSL 스타일시트와 컴포넌트 객체의 사용이다. XSL은 XML 문서의 외형을 만들어 내는 매우 융통성 있는 도구로써, XML 파일과 XSL 파일이 결합하여 HTML 파일을 만들 수 있으며 데이터를 정렬하거나 해당되는 내용을 찾기 위해 패턴 매칭을 할 수 있다. COM+ 컴포넌트 객체의 사용은 ASP 프로그램에서 서버 엔진의 자세한 구현 사항을 숨길 수 있으므로 프로그램 작성을 간단히 하고 이의 해독력을 향상시킨다.

COM+ 컴포넌트 객체에는 전체적인 프로세스를 관리하

는 clsPlanManager라고 하는 하나의 클래스를 가지고 있다. 이 클래스는 다음과 같이 세 개의 메소드로 구성된다.

- getPlan( ) 질의 요청에 대해 패턴 일치되는 서비스 리스트를 포함하고 있는 XML을 리턴하는 메소드
- getPlanData( ) 리스트 중 한 항목에 대해 수집된 입력 파일을 포함하고 있는 XML을 리턴하는 메소드
- executePlan( ) 실행 요청에 대해 관련 서비스를 실행하는 메소드

clsPlanManager 중에서 executePlan( ) 메소드는 확장 가능하도록 만들 필요가 있다. 이 메소드의 역할은 개별 항목을 기술하는 XML 파일을 입력으로 받아서 그 서비스의 타입(예, Script, Exec, External Service)을 결정하고 이를 위한 적합한 실행 환경을 제공하는 것이다. 새로운 서비스 타입(예, DB Service)을 쉽게 추가할 수 있도록 각 타입마다 독립된 클래스를 제공하고 이들은 일관된 인터페이스를 제공해야 한다. 이를 위해 ifcService라는 인터페이스를 만들었다. 이의 구성은 다음과 같다.

- Implements ifcService
- ifcService\_getService( ) 서비스를 가져오는 메소드
- ifcService\_startService( ) 서비스를 시작하는 메소드
- ifcService\_completeService( ) 서비스가 끝났는지 확인하는 메소드
- ifcService\_stopService( ) 서비스를 끝내는 메소드

본 연구에서는 데이터 교환을 위해 XML을 사용함으로써 플랫폼 독립성을 지원하고 서버와 클라이언트 쪽에 유연성을 향상시킨다. 클라이언트에서는 서버가 어떤 데이터베이스 시스템이나 파일 시스템을 사용하더라도 자유롭게 응용 프로그램을 구성할 수 있다. 서버쪽에서도 클라이언트가 어떤 프로그램을 수정하더라도 서버 프로그램을 수정할 필요가 없으므로 시스템 독립성을 유지할 수 있다.

### 5. 실험

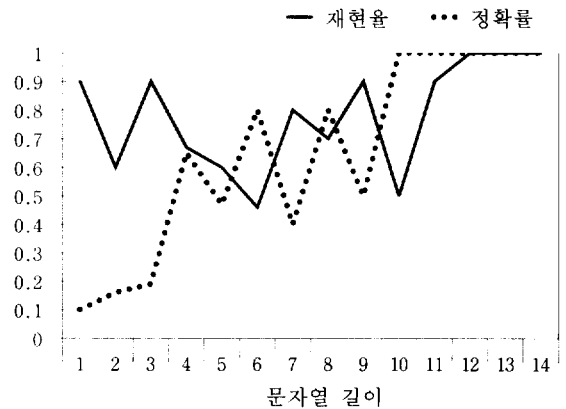
3.3에서 제안한 서비스 검색 알고리즘의 실험을 위해 Princeton 대학[17]에서 제공하고 있는 비선형 최적 모델(nonlinear optimization models)을 이용하였다. 이 사이트에서는 약 900개 정도의 GAMS 모델을 제공하고 있는데 이들 중 "Cute Set" 디렉토리에 있는 306개의 프로그램을 SSM 그래프로 변환하고 이에 대한 정렬된 문자열을 DBMS에 저장하였다.

평가 방법은 정보검색에서 가장 보편적으로 사용되고 있는 재현율(recall)과 정확률(precision)을 사용한다. 본 실험의 경우는 서비스 집합과 질의 모두 SSM 문자열이 되기 때문에 재현율과 정확률은 다음과 같이 정의된다.

$$\text{재현율} = \frac{\text{검색된 서비스들 중 같은 부류에 속하는 서비스의 수}}{\text{같은 부류에 속하는 서비스의 수}}$$

$$\text{정확률} = \frac{\text{검색된 서비스들 중 같은 부류에 속하는 서비스의 수}}{\text{검색된 서비스의 수}}$$

검색 대상이 되는 문자열과 질의 문자열을 따로 마련하지 않고, 문자열 하나씩 질의로 사용하고 나머지 모든 문자열은 검색 대상 집합이 된다. 이렇게 모든 문자열에 대해 재현율과 정확률을 측정하고, 이들을 문자열 길이에 따라 평균을 내었다. 이에 대한 결과는 (그림 11)과 같다.



(그림 11) 실험 결과

재현율과 정확률은 모두 높을수록 성능이 좋다. 하지만 이들은 서로 반비례의 관계가 있어 한쪽을 높이면 다른 한쪽이 내려가는 것이 보통이다. 실험 결과 정확률에 비해 재현율이 비교적 높는데 이는 같은 부류에 속하는 프로그램들은 거의 같은 SSM 구조를 가지게 되며, 이 문자열을 질의로 사용했기 때문에 같은 부류에 속하는 서비스들이 거의 검색되기 때문이다. 또한 문자열 길이가 짧을 때 매우 낮은 정확률을 나타내고 있는데 이는 Cute Set 서비스들이 비교적 간단하여 전체의 29%가 2-3개의 문자열로 구성되어 있기 때문에 이에 대한 검색은 많은 검색 결과 수에 비해 정확도가 떨어지기 때문이다. 그렇지만 문자열 길이가 길어질수록 정확률이 높아지는데 이는 다른 서비스들 끼리 같은 문자열 패턴을 가질 확률이 매우 낮아지기 때문이다.

실험에 의하면 문자열 길이가 5개 이상 되는 서비스들에는 적절한 재현율과 정확률을 나타내고 있다. 문제는 공급자들은 GAMS 문법에 익숙하고 그들의 서비스를 쉽게 SSM으로 변환할 수 있지만 수요자는 GAMS 문법과 SSM에 익숙하지 못한 경우가 있을 수 있다는데 있다. 이를 위해 SSM 변환 마법사(wizard)를 제공하여 초보자라도 쉽게 SSM 그래프를 생성할 수 있도록 지원해 줄 필요가 있으며, 정확한 SSM을 생성하지 못하였더라도 일치도가 높은 순위부터 나열해 주는 유사 일치도 측정 모델을 지원할 필요가 있다.

### 6. 결론

본 논문에서는 인터넷 마켓플레이스를 위한 XML 기반 계획 에이전트를 설계하고 구현하였다. 이를 위해 관련 분



야인 다중데이터베이스 시스템과 워크플로우 관리시스템의 특징을 비교 분석하고 계획 에이전트의 기능과 역할을 기술하였다. 설계된 계획 에이전트는 COM+에서 작동되는 하나의 미들웨어로써 구현되었다. 클라이언트/서버 프로그램을 위해 ASP를 사용하였고 ASP와 계획 에이전트 간 자료 교환을 위해 XML을 사용하였다. 이 시스템은 실제 운영되고 있는 MMM으로부터 메타 정보를 제공받아 실험을 통해 그 적용 가능성을 확인하였다.

본 연구에서 제안된 서비스 검색 알고리즘의 실험을 위해 Princeton 대학에서 인터넷으로 제공하고 있는 비선형 최적 모델을 사용하여 성능 분석을 수행하였다. 실험 결과 어느 정도 복잡한 서비스들에게는 적절한 재현율과 정확률을 제공하고 있다. 하지만 성공적인 인터넷 마켓플레이스를 구축하기 위해서는 본 연구에서 취급하지 못한 많은 기술들을 해결해야만 한다. 특히 동적 인터넷 환경 하에서의 스키마 통합 문제라든가 계획 에이전트에서 SSM 그래프를 쉽게 생성할 수 있는 변환 마법사, 그리고 유사 일치도를 우선순위로 나열해 주는 유사치 측정 메카니즘 등에 대한 연구가 향후 필요하다.

**부 록**

\$ TITLE A TRANSPORTATION PROBLEM (TRANSPORT, SEQ=1)

\$ OFFUPPER

- \* This problem finds a least cost shipping schedule that
- \* meets requirements at markets and supplies at
- \* factories
- \* References :
- \* Dnatzig, G B., Linear Programming and Extensions
- \* Princeton University Press, Princeton,
- \* New Jersey, 1963, Chapter 3-3

SETS

I plants /SEATTLE, SAN-DIEGO /

J markets /NEW-YORK, CHICAGO, TOPEKA / ;

PARAMETERS

A(I) supply of plant i in cases

/SEATTLE 350  
SAN-DIEGO 600 /

B(J) demand at market j in cases

/NEW-YORK 325  
CHICAGO 300  
TOPEKA 275 / ;

TABLE D(I, J) distance in thousands of miles

NEW-YORK CHICAGO TOPEKA

SEATTLE 2.5 1.7 1.8

SAN-DIEGO 2.5 1.8 1.4 ;

SCALAR F freight in dollars per case per thousand miles /90/ ;

PARAMETER C(I, J) transport cost in thousands of dollars per case ;

$$C(I, J) = F * D(I, J) / 1000 ;$$

VARIABLES

X(I, J) shipment quantities in cases

Z total transportation costs in thousands of dollars ;

POSITIVE VARIABLE X ;

EQUATIONS

COST define objective function

SUPPLY(I) observe supply limit at plant i

DEMAND(J) satisfy demand at market j ;

COST ..  $Z = E = \text{SUM}((I, J), C(I, J)*X(I, J)) ;$

SUPPLY(I) ..  $\text{SUM}(I, X(I, J)) = L = A(I) ;$

DEMAND(J) ..  $\text{SUM}(I, X(I, J)) = G = B(J) ;$

MODEL TRANSPORT /ALL/ ;

option lp = sciconic ;

SOLVE TRANSPORT USING LP MINIMIZING Z ;

DISPLAY X.L, X.M ;

**참 고 문 헌**

- [1] H. A. Jacobsen and O. Guenther, "Middleware for Software Leasing over the Internet," ACM Conf. on Electronic Commerce (EC-99), Denver, Colorado, USA, November 1999.
- [2] O. Guether and R. Mueller, "From GISystem to GIServices : Spatial Computing on the Internet Marketplace," Interoperability in Geographic Information Systems, 1998.
- [3] D. J. Abel, "Spatial Internet Marketplaces : A Grand Challenge?" Proc. 5th Int. Symposium on Spatial Databases (SSD '97), 1262 LNCS, Springer-Verlag, 1977.
- [4] H. K. Bhargava, R. Krishnan, S. Roehrig, M. Casey, D. Kaplan, R. Mueller, "Model Management in Electronic Markets for Decision Technologies : A Software Agent Approach," Proc. 13th Int. Conf. on System Sciences, Maui, Hawaii, January 1997.
- [5] O. Guenther, R. Koerstein, R. Krishnan, "The MMM Project : Access to Algorithms via WWW," Proc. 3th Int. World-Wide Web Conf., Darmstadt, Germany, April 1995
- [6] D. J. Abel, VTowards Spatial Internet Marketplaces," GeoInformatica, . Gaede, K. L. Taylor, X. Zhou, "SMART : 3(2), pp.141-164, 1999.

[7] 이경하, 이강찬, 이규철, "XML 프로그래밍", 정보과학회지, 제 18권 제4호, pp.4-12, 2000년 4월.

[8] A. P. Sheth and J. A. Larson, "Federated Database Systems for Managing Distributed Heterogeneous, and Autonomous Databases," Computing Surveys, 22(3), pp.182-236, 1990.

[9] WfMC, "WfMC(Workflow Management Coalition) Standard Documents," Technical Report, Workflow Management Coalition, November 1998.

[10] 신동일, 신동규, "워크플로우 관리 시스템의 설계 및 구현", 정보처리논문지, 제7권 제5호, pp.1609-1619, 2000년 5월.

[11] NIST, GAMS Guide to Available Mathematical Software, <http://gams.nist.gov>, 1998.

[12] R. Mueller and S. Mueller, "Retrieval of Service Descriptions Using Structured Service Models," Proc. WITS 2000, Brisbane, Australia, 2000.

[13] A. M. Geoffrion, "The Formal Aspects of Structured Modeling," Operation Research, 37(1), pp.30-51, 1989.

[14] K. Chari and T. K. Sen, "An Implementation of a Graph-based Modeling System for Structured Modeling (GAMS/SM)," Decision Support Systems 22, pp.103-120, 1998.

[15] K. Gayer and V. Graede, "Structured Documents for Representing Services in Internet Marketplaces," Technical Report

ISS-55, Humboldt University, March 1998.

[16] S. Mohr, Designing Distributed Applications with XML, ASP, IE5, LDAP, and MSMQ, Wrox Press Ltd., 1999.

[17] R. Vanderbei, Nonlinear Optimization Models, Princeton University, <http://www.sor.princeton.edu/~rvdb/ampl/nlmodels/index.html>, 2000.



## 이용주

e-mail : yongju@sangju.ac.kr

1983년 울산대학교 산업공학과(공학사)

1985년 한국과학기술원 산업공학과  
(공학석사)

1997년 한국과학기술원 정보및통신공학과  
컴퓨터공학전공(공학박사)

1985년~1989년 시스템공학연구소 연구원

1987년~1988년 일본 IBM TRL 연구소 연구원

1989년~1994년 삼보컴퓨터 근무

1997년~1998년 한국과학기술원 Post Doc.

1998년~현재 상주대학교 컴퓨터공학부 조교수

관심분야 : 웹 데이터베이스, 정보검색, 공간 데이터베이스