

# 전자상거래를 위한 정책지향 매칭 에이전트 시스템의 설계 및 구현

황 병 연<sup>†</sup> · 박 성 철<sup>\*\*</sup>

## 요 약

최근 인터넷 솔루션 시장은 전자상거래의 필요성과 효과에 대한 논의를 벗어나 전자상거래 내에서의 경쟁에 초점이 맞추어져 있다. 본 논문에서는 전자상거래 시장의 활성화에 기여할 수 있는 정책지향 매칭 에이전트(policy-oriented matching agent) 시스템을 제안한다. 제안된 정책지향적 솔루션은 정책을 객체화하였기 때문에 다양한 프로모션을 구현할 수 있으며, 기존의 룰 기반 시스템의 장점에 정책이 실시되는 공간 개념(release post)을 추가하였기 때문에 정책의 수립, 실시, 추적, 평가 등을 일관되게 처리할 수 있다. 정책지향 솔루션을 구현함에 있어 전자상거래를 위한 기초 플랫폼에 적합한 컴포넌트 기반 구조를 채택하고, 기존의 여타 시스템에 연동되어 마케팅 활동을 쉽게 지원할 수 있도록 한다. 또한 정책 담당자가 정책을 직관적으로 편집할 수 있는 인터페이스를 갖도록 한다.

## Design and Implementation of Policy-oriented Matching Agent System for Electronic Commerce

Byung-Yeon Hwang<sup>†</sup> · Sung-cheol Park<sup>\*\*</sup>

## ABSTRACT

Recently, Internet solution market is focused on competition in Electronic Commerce (EC) rather than necessity and effectiveness of EC. In this paper, we propose a policy-oriented matching agent system that makes EC market active. We can pursue diverse promotion since policies are objects, and we can formulate, carry out, pursue, and evaluate a policy consistently on the proposed solution since our solution is integrated into merits of rule-based system and operation space (release post). There are three important points of the implementation. It is component-based structure that fits to be a basic platform for EC. It is connected with other systems easily, so it can help marketing. Marketing managers can edit marketing policies easily through the system, because it has an intuitive interface.

키워드 : 정책지향 매칭 에이전트(policy-oriented matching agent), 전자상거래(electronic commerce)

### 1. 서 론

최근 인터넷 솔루션 시장은 전자상거래의 필요성과 효과에 대한 논의를 벗어나 전자상거래 내에서의 경쟁에 초점이 맞추어져 있다[1-3]. 솔루션에 있어서도 초기에는 기본적인 플랫폼을 담당한 서버나 보안 솔루션을 중시했던 반면 근래에는 고객에게 양질의 서비스를 제공하여 보다 많은 이익을 얻을 수 있는 솔루션이 부각되고 있다. 즉, 새로운 고객을 유치하고 한 번 고객이 되면 그 관계를 오래도록 지속시킬 수 있는 정책적 활동에 대한 필요성을 느끼고 있다. 이러한 인식에 따라 효율적인 매칭 에이전트 시스템 개발의 중요성이 더욱 부각되고 있다[4-6].

본 논문에서는 시스템 설계의 관점에서 정책지향 솔루션의 개념과 구조를 설정하고, 국내에서 시장 점유율이 높은 솔루션에 대해서 '마케팅 지원' 기능을 중심으로 하여 비교 분석하였다. 또한 설계된 정책지향 솔루션을 Window NT 상에서 C++ 언어를 이용하여 구현하였다. 구현 시 전자상거래를 위한 기초 플랫폼에 적합한 구조를 갖추기 위해 컴포넌트 기반 구조를 채택하였고, 기존의 여타 시스템에 쉽게 연동되어 마케팅 활동을 지원할 수 있도록 하였으며, 정책 담당자가 정책을 직관적으로 편집할 수 있는 인터페이스를 갖도록 하였다.

정책지향 매칭 에이전트는 기존의 룰 기반 매칭 에이전트 솔루션들에 대한 분석을 통해서 많은 부분을 흡수하여 발전시킨 개념이다. 특히 BroadVision의 One-To-One(www.broadvision.com)과 ATG의 Dynamo(www.atg.com)를 룰 기반 매칭 에이전트의 대표적인 사례로 보고 분석했다. 이 과정의

※ 본 연구는 2001년도 가톨릭대학교 교비연구비의 지원으로 이루어졌음.

† 종신회원 : 가톨릭대학교 컴퓨터전자공학부 교수

\*\* 준 회원 : 가톨릭대학교 컴퓨터공학과

논문접수 : 2001년 1월 9일, 심사완료 : 2001년 9월 25일

결과로서 룰 기반 매칭 에이전트에서는 정책지향성이 부족하다는 것을 지적하고 이를 보완하기 위한 새로운 시스템 구조를 제시하였으며 룰의 개념으로 정책을 표현할 수 있도록 확장, 응용하였다.

정책은 비즈니스 룰에 의해서 그 의미와 목적이 표현되는 추상적 정책이며, 이들 정책은 실시 공간에 특정한 옵션(release option)으로 할당되어 실시되어진다. 정책지향 솔루션은 기존의 룰 기반 시스템의 장점과 정책이 실시되는 공간 개념(Release Post)이 결합되어 있기 때문에 정책의 수립, 실시, 추적, 평가 등을 일관되게 처리할 수 있는 솔루션이다.

본 논문의 구성은 다음과 같다. 2장에서는 기본 기술로서의 룰 기반 매칭에 대해 살펴보고, 3장에서는 본 논문에서 제안하는 정책지향 매칭 에이전트 솔루션의 논리적 구조 및 장점을 제시한다. 4장에서는 정책지향 솔루션의 구현으로서 시스템 개요와 구조 그리고 주요 구성요소를 기술하고, 5장에서는 정책지향 솔루션과 기존의 룰 기반 솔루션을 적용가능 예를 중심으로 비교한다. 끝으로 6장에서 결론을 맺는다.

## 2. 기본 기술로서의 룰 기반 매칭

정책지향 룰 기반 매칭 에이전트의 성격을 정리하기 위해서 우선 마케팅 활동의 일반적인 프로세스부터 살펴보자. 전형적 마케팅 활동의 프로세스는 마케팅 정책(전략)의 수립, 실행, 분석이라는 세 가지로 나눌 수 있다[7]. 마케팅 정책의 수립은 기존의 데이터를 이용하거나, 새로운 환경을 고려하여 보다 나은 영업 촉진정책(promotion)을 뽑아내는 과정이다. 마케팅 정책의 실행은 이렇게 수립된 정책을 구현할 수 있는 각각의 채널들을 통해서 실시하는 것을 의미한다. 여기서 채널이라고 하는 것은 과거 데이터베이스 마케팅의 경우 주로 DM(direct mail), 전화 등이었으나, 현재 전자상거래에서는 웹 페이지나 배너 광고 등이 될 수 있다. 물론 동일한 채널에서도 프로모션의 내용에 따라서 가격 할인일 수도 있고, 신상품 기획 등 각각 다른 질을 가질 수도 있다. 마케팅 정책의 분석은 실시된 마케팅 정책의 결과 및 효과를 양적, 질적으로 분석해 내는 과정을 의미하며, 이는 새로운 정책수립으로 피드백 되는 중요한 프로세스이다.

각 고객에게 적합한 제품과 서비스를 제공해 주는 개별화 작업은 매칭 기술에 의해 구현된다. 매칭 기술은 에이전트에 의한 푸시와 사용자의 풀로 일차적으로 구분된다. 에이전트 푸시기술은 다시 통제 주제에 따라 매칭 에이전트가 주도하는 룰 기반 매칭, 피드백 및 학습, 사용자가 주제인 매칭 에이전트, 그룹평가, 그리고 필터링으로 나뉜다. 사용자 풀기술도 속성기반 조건검색, 전문검색, 메뉴선택, 그리고 playback 기술로 다시 분류된다. 매칭 기술 중 가장 효과적인 것이 룰 기반 매칭이다[8]. 사용자 풀기술은 고객에게 그가 정확히 찾고자 하는 것이 무엇인지 알아야 한다는 부담을 준다. 하지

만 에이전트 푸시기술은 고객에게 이러한 부담을 주지 않는다. 또한 사이트를 관리하는 입장에서 보면 에이전트 푸시기술은 사용자들과의 일대일 관계를 유지하고 상품과 서비스들을 크로스 셀링(cross selling)할 수 있다는 측면에서 가치를 측정할 수 없을 정도이다[9, 10]. 따라서 기존의 매칭 기술 중 정책 관리자의 입장에서 가장 효과적으로 정책을 실현할 수 있는 기술은 룰 기반 매칭 기술이라고 할 수 있다. 그러나 룰 기반 매칭 기술의 경우 정책결정과정과의 유기성 부분에서 정책의 수립, 실행, 분석에 대한 별도의 데이터 수집 기능이 취약하고, 룰의 정책구현 능력이 미흡하다는 문제점이 있다. 이 문제점의 원인과 이를 해결하기 위해 본 연구에서 개발한 개념인 정책지향 매칭 에이전트를 3장에서 살펴보겠다.

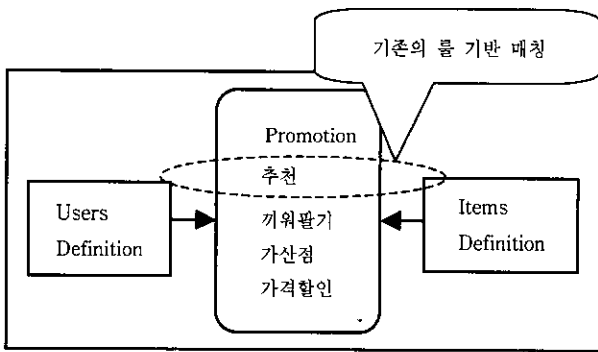
## 3. 정책지향 매칭 에이전트

시스템의 정책지향성이란 마케팅 정책을 수립하고 실시, 추적, 분석하는 행동들이 쉽고 집중적으로 이루어지는 정도를 말한다. 정책지향성이 강한 솔루션일수록 시스템의 전체적인 동작을 정책이라는 개념을 중심으로 파악하기도 쉽고, 변화를 가하기도 간단하다. 정책지향성을 구현하기 위해서는 우선 시스템 설계의 관점에서 볼 때, '정책'이 하나의 객체로서 존재해야 한다. 가격할인이라는 익숙한 마케팅 정책의 예를 통해 정책이라는 객체가 가지는 속성과 그 작용을 설명하면 다음과 같다. 가격할인은 해당 사이트에 들어오는 모든 사람들에게 적용될 수도 있지만 우수 고객들만을 대상으로 할 수도 있다. 그리고 모든 상품들을 할인할 수도 있으며 특정 상품들에만 적용할 수도 있다. 어쨌든 할인이란 정책은 정책을 실시할 사용자 대상선정 조건과 그에 따른 대응을 특수하게 구체화한 것이다. 주의할 것은 대상선정 조건은 사용자나 아이템에 모두 해당된다는 점이다.

본 연구에서는 정책지향 매칭 에이전트 구현을 위해 기존의 룰 기반 매칭 시스템을 이용하였다. 룰 기반 매칭 에이전트는 비즈니스 관리자가 정한 비즈니스 룰에 따라서 특정 콘텐츠를 특정 사용자에게 매칭시켜주는 시스템이다. 이 에이전트는 자동화된 에이전트가 아직까지 충족시켜 줄 수 없는 다양한 비즈니스 룰을 지원할 수 있기 때문에 현재 마케팅 지원기술로서 가장 널리 이용되고 있다[11, 12]. 그러나 기존의 룰 기반 매칭 에이전트들은 마케팅 정책의 라이프 사이클 중 구현에만 중점을 두었기 때문에 정책의 수립 및 평가에는 취약한 편이다. 이와 같은 한계는 기존의 룰이 대상을 선정하는 조건에 따라 아이템을 추천하도록 하는 두 단계로 구성된 것에 기인한다. 특히 두 번째 단계인 아이템 추천의 경우 추천이라는 행동에 강조를 함으로써 가격의 할인이나 끼워 팔기, 묶어 팔기 등 다양한 프로모션 정책을 표현하기 힘들다. 이런 문제를 해결하려면 정책이 사용자와 대상의 선정 그리고 그에 따른 대응(프로모션)으로 분명히 나누어져야

한다.

사용자 정의는 정책이 적용될 사용자를 구분해 내기 위한 것이며, 이는 전통적인 데이터베이스 마케팅에서 고객 세그먼트를 정의하는 것과 동일하다. 아이템에 대한 정의는 해당 마케팅 활동이 적용될 아이템들을 구분해내기 위해 필요하다. 이 두개의 대상에 대해서 어떻게 결합시킬 것인가 하는 것이 바로 프로모션이다. 프로모션은 가격의 할인율일 수도 있고 안내 메시지일 수도 있으며, 보너스 포인트 증가일 수도 있고 이 밖에도 여러 가지가 있을 수 있다. 이것은 정책을 표시하는 기본적인 모델이며 대상 사용자 정의, 대상 아이템 정의, 프로모션 정의에 물을 사용한다는 것이다. 이때 물이란 시스템 상에서 자체로 정의되는 문법구조를 가지게 될 것이다. 이렇게 본 논문에서 제안된 정책을 표현하게 되면, (그림 1)과 같이 정책지향 구조는 기존의 물 기반 매칭을 포함하는 상위 집합이 된다.



(그림 1) 정책지향 매칭의 개념

정책지향 솔루션에서 물에 의해 표현된 정책은 특정한 실시 옵션을 가지고 특정공간에서 실시된다. 이 세 가지의 개념을 보다 분명히 하면서 그 장점을 정리해보자. 실시공간이나 개념을 명확히 분리하는 것은 정책을 추적하고 평가하기 위해서 절대적으로 필요하다. 실시공간은 특정 웹 페이지처럼 고유한 공간이다. 이 공간이 시스템에서 따로 파악되기 때문에 어떤 정책이 어느 공간에서 실행되고 있는지, 동일한 정책이라도 공간에 따라 어떻게 다른지 등의 문제를 파악해 낼 수 있게 된다. 한편 실시 옵션은 정책과 정책공간이 만나는 방식을 정의하는 것으로서, 특히 시간적 제약을 표시하거나 여러 개의 정책을 한 공간에 섞어가며 적용하는 등의 기능을 할 수 있도록 해준다.

#### 4. 정책지향 솔루션의 구현

##### 4.1 목적 및 요구사항

이제까지 정책지향성과 정책지향 솔루션의 개념에 대해서는 논의했다면, 이제는 그 개념을 적용한 소프트웨어적인 설계의 구현에 대해 논의한다. 항상 그러하듯이 소프트웨어의 설

계는 분석에 따른 핵심적인 개념을 기반으로 하지만, 이외에도 많은 다른 요인들이 참고 될 수밖에 없으며 또 참조해야만 한다. 우선, 시장에서의 경쟁력 또는 사업화 가능성이란 점을 고려해야 하며, 시스템이 안정적으로 운영되면서도 성능 면에서 만족스러워야 한다는 것도 중요하게 고려할 부분이다. 구현에서 설정한 요구사항은 다음과 같이 정리할 수 있다.

첫째, 전자상거래를 위한 기초 플랫폼에 적합한 구조를 갖추어야 한다. 중소형 전자상거래 사이트의 경우에 이 솔루션만으로도 사이트를 운영하는 것이 가능해야 하기 때문에 전방위(front-end), 후방위(back-end)에 걸친 영역에 대해서 추가기능의 설치만으로 쉽게 처리할 수 있는 구조를 지니고 있어야 한다. 이를 위해서는 컴포넌트 기반 구조(component based architecture)의 채택이 필수적이다.

둘째, 여타의 시스템에 연동되어 마케팅 활동을 쉽게 지원할 수 있어야 한다. 솔루션은 토탈 솔루션으로서 기능하는 한편, 정책지향성을 구현하기 위한 ‘추가 기능적 성격’(addon ability)을 가져야 하는데, 이는 무엇보다 시장에서의 경쟁력, 즉 사업성을 고려한 것이다. 현재 국내시장 및 해외시장에서는 토탈 솔루션 시장의 주인들이 거의 모습을 모두 드러낸 상태이다. 그러나 전자상거래 시장에서 CRM(customer relationship management)과 같은 특화된 솔루션[13], 특히 마케팅분야에 특화된 솔루션 시장은 이제 열리고 있으므로, 이와 같은 신규시장에 효과적으로 대응하고, 또 기존 상용사이트에도 진출하기 위해서는 추가 기능적 성격을 가지는 것이 매우 중요하다. 추가 기능적이란 말은 다른 커머스 플랫폼에서도 쉽게 연동되어 동작한다는 의미다. 예를 들어 Oracle, MS, IBM의 플랫폼에 연동되어 쉽게 동작할 수 있어야 된다는 것이다. 또 한편으로는 고객의 기존 시스템에도 쉽게 연동될 수 있어야 한다.

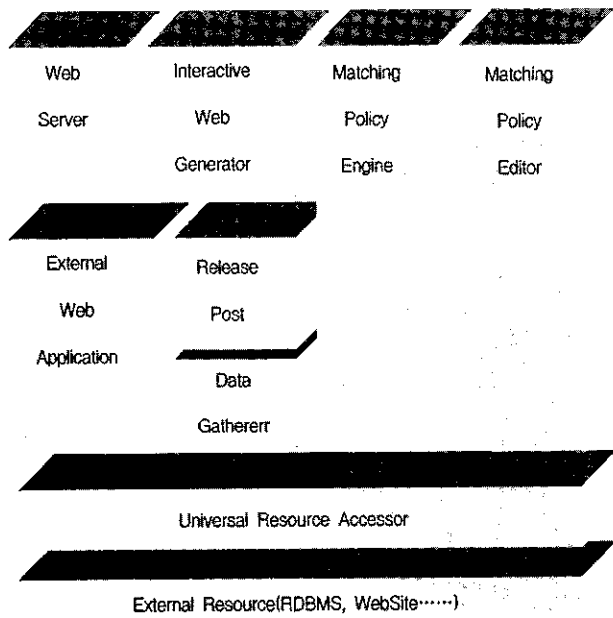
셋째, 마케팅 활동의 수립, 실시, 추적, 분석에 따르는 일련의 과정을 시스템이 온전히 받아들일 수 있는 정책지향 솔루션 구조로 구현되어야 한다. 이는 개발 솔루션의 기능상의 주요 특징이며, 타 제품과 비교했을 때 차별성의 기본이 되고, 따라서 경쟁력의 결정 요소가 된다고 볼 수 있다.

넷째, 정책 담당자가 정책을 직관적으로 편집할 수 있는 인터페이스를 가져야 한다. 기술자인 정책 담당자가 직관적으로 정책을 관리하는 것은 비용을 절감하고, 적기에 적절한 정책을 구사할 수 있다는 점에서 매우 중요한 요소이다.

##### 4.2 시스템의 구조

개발된 시스템은 정책지향 매칭 에이전트(Matching Policy Engine), Release Post, Data Gatherer, URA(Universal Resource Accessor)를 중심으로 각 유스케이스에 따라 다른 모듈이 자신의 역할을 수행하는 중앙 집중식 형태를 갖는다(그림 2). 웹 어플리케이션 유스케이스일 때에는 정책지향 매칭

에이전트에 생성되어있는 정책이 웹 어플리케이션과 연동되는 모듈인 Release Post에서 실시되고 그에 따른 사용자의 반응이나 결과를 Release Post가 정책지향 매칭 에이전트로 보내준다. 이 때, Data Gatherer는 특정 정책에 국한되지 않는 일반적인 사용자 이용 데이터인 웹 traffic정보를 모아서 정책지향 매칭 에이전트에 보내준다. 정책 생성 및 실시 유스케이스에서는 정책 관리자가 정책 관리 툴(Matching Policy Editor)을 통해 정책지향 매칭 에이전트에 정책을 수립하고 해당 Release Post를 생성하거나, 정책 및 Release Post를 변경 또는 삭제할 수 있다. 독립 실행형 어플리케이션 유스케이스 일 때에는 별도의 웹 어플리케이션이 없으므로 Release Post와 Data Gatherer가 IWG(Interactive Web Generator)와 연동되어 사용자에게 대한 정책의 실시와 사용자 이용 정보 수집이 이루어진다. 리포팅 유스케이스의 경우에는 정책지향 매칭 에이전트가 정책 분석가에게 Release Post 별 추적 자료와 정책별 추적 자료를 제공해준다.



(그림 2) 시스템 구조

### 4.3 주요 구성요소

#### 4.3.1 Release Post

Release Post는 정책이 실시되는 곳에 설치되는 소프트웨어 모듈이다. 모든 Release Post는 정책 담당자에 의해서 개설되고 권한이 지정되며 삭제된다. Release Post는 웹 서버나 타 어플리케이션에 장착될 수 있으며, 심지어 CTI(computer telephone integration), TV, Kiosk 등 인터넷이 아닌 다른 환경에도 장착되어 정책 서비스가 그 시스템에서 이용될 수 있도록 한다. 따라서 Release Post는 그것이 장착되는 시스템에 따라서 다양한 형태로 제작될 수 있다.

#### 4.3.2 정책지향 매칭 에이전트(Matching Policy Engine)

정책지향 매칭 에이전트는 정책지향 솔루션의 핵심 엔진으로서, 정책관리자의 정책 정의 텍스트를 분석하고 이해하며 관리한다. 그리고 이 에이전트는 Release Post의 요구에 의해서 그 Release Post에 할당되어진 정책을 실행하고 결과를 Release Post에게로 전송하는 역할을 담당한다. 정책지향 매칭 에이전트가 참조하는 기초 데이터는 모두 URA(Universal Resource Accessor)와의 표준적인 API로 통하게 되어 있다. 정책지향 매칭 에이전트의 알고리즘은 다음과 같다.

```
public class matchingPolicyManager { // Matching Policy Editor의 요청에 의한 정책의 생성, 변경 및 삭제
    .....
    public void setPolicy(String targetCustomer, String targetItem, String Promotion)
        // 정책 관리 툴로부터 정책 정보를 넘겨 받음. 에러확인, 스크립트 컴파일
    {
        .....
    }
    public String getCustomer() // 정책 관리 툴에 고객정보를 넘겨줌
    {
        return targetCustomer ;
    }
    public String getItem() // 정책 관리 툴에 상품정보를 넘겨줌
    {
        return targetItem ;
    }
    public String getPromotion() // 정책 관리 툴에 프로모션 정보를 넘겨줌
    {
        return targetPromotion ;
    }
    public void matchingPolicyManage(String policyName, String editMode) // 매칭 정책 관리
    {
        if(editMode == "show") // 해당 Policy정보를 정책 관리 툴에 넘겨줄 준비
        {
            showPolicy(policyName) ;
        }
        elseif(editMode == "create") // 새 정책 수립
        {
            if(checkCustomer(policyName) // 기존 정책의 이름과 중복되지 않으면
            {
                addNewPolicy(policyName) ;
                // 입력된 정책 이름으로 대상 고객, 대상 아이템, 프로모션을 정책 관리 툴에서 받아서 툴 리스트에 등록시킴
            }
            else
            {
                error() ;
            }
        }
        elseif(editMode == "modify") // 기존 정책 수정
        {
            modifyPolicy(policyName) ;
        }
        elseif(editMode == "delete") // 해당 정책 삭제
        {
            deletePolicy(policyName) ;
        }
    }
}
```

```

}

public void setReleasePost(String targetGroup, String targetRelease
Post, String targetPolicyName)
//정책 관리 톨로부터 ReleasePost 관련 정보를 받아옴
{
.....
}

public String getGroup() // 정책 관리 톨에 group정보를 넘겨줌
{
return targetGroup ;
}

public String getRelease Post() // 정책 관리 톨에 ReleasePost 정
보를 넘겨줌
{
return targetReleasePost ;
}

public String getPolicyName() // 정책 관리 톨에 해당 정책 이름
을 넘겨줌
{
return targetPolicyName ;
}

public void releasePostManage(String editMode) // Release Post
관리
{
if(editMode == "new")
{
creatReleasePost(targetGroup, targetReleasePost, targetPolicy
Name) ;
// ReleasePost를 특정 Release Post Group에 생성하고 실행
정책을 선택함
}elseif(editMode == "show")
{
showReleasePost(targetGroup, targetReleasePost) ;
// 해당 Release Post의 정책들을 정책 관리 톨로 전송
}
elseif(editMode == "modify")
{
modifyReleasePost(targetGroup, targetReleasePost, targetPolicy
Name) ;
// 해당 Release Post를 수정함
}
elseif(editMode=="delete")
{
deleteReleasePost(targetGroup, targetReleasePost, targetPolicy
Name) ;
// 해당 Release Post를 지움
}
else
{
errorMessageOutAndErrorCheck() ;
}
}
}

public void matchingPolicyProcessor(user, userEvent) // 사용자 이벤
트 발생에 의한 정책의 실시 및 이용정보 수집
{
String targetReleasePost = selectReleasePost(user, userEvent) ;
// 사용자 및 사용자 이벤트로부터 해당 Release Post 판별
int presentRuleNumber = 0 ;
While(endOfTheRuleList(presentRuleNumber)) // 해당 Release Post
상에 적용되는 모든 톨을 실시하고 톨 리스트의 모든 톨을 확인했
으면 while문을 끝냄

```

```

{
if(targetRule(targetReleasePost, user, userEvent, presentRule
Number)) // 톨 리스트 중 현재 확인중인 톨이 실시 대상 톨이면
{
processTargetRule(user, userEvent, presentRuleNumber) ;
//사용자를 식별하고 사용자 프로필 정보와 이익을 검사하여 톨
을 평가하고, 결과 매칭 콘텐츠 및 콘텐츠 속성에 가중치 적용
}
presentRuleNumber++ ; // 다음 톨로 이동
}
sendResultToReleasePost() ; // 결과 매칭 콘텐츠를 사용자에게 제
시하기 위해 처리된 결과를 Release
Post에 보냄
}
}

```

#### 4.3.3 IWG(Interactive Web Generator)

독립 실행형 웹 어플리케이션으로 이용될 경우에 필수적인 동적 HTML생성 기능을 가지고 있는 CORBA서버이다. 이 서버는 자체 스크립트 엔진을 가지고 있고, 컴포넌트 기반 구조를 채택하고 있기 때문에 쉽게 웹 페이지를 생성하고, 기능을 확장할 수 있도록 해준다.

스크립트 엔진은 최대한 자바 스크립트와 동일한 문법을 가지도록 하여, 웹 페이지 개발자들이 쉽게 적용할 수 있도록 했다. 이러한 독자적인 스크립트 언어를 구현한 것은 독립 실행형 어플리케이션으로 응용될 때, 다른 여타의 시스템 도움이 없이도 동적 HTML 생성을 지원하기 위함이다. 그리고 이 스크립트 언어는 새로운 컴포넌트를 인식할 수 있어 추가 기능의 설치 및 확장이 용이하다는 장점이 있다.

#### 4.3.4 URA(Universal Resource Accessor)

URA는 정책지향 매칭 에이전트가 요구하는 표준적인 API를 통해서 다양한 형태의 자료를 처리하는 CORBA서버이다. URA에 대한 데이터 요구는 URL을 응용한 표준적 질의 방식을 채택하고 있다.

예를 들어, 정책지향 매칭 에이전트 엔진은 "MSSQL : user : pwd@host.co.kr/pub/table?attribute = 'j'"라는 요구를 URA에게 보낼 수 있다. 이때 정책지향 매칭 에이전트는 MSSQL이라는 프로토콜을 통해서 해당 자료를 처리할 URA를 찾아내게 되고, 나머지 요구를 해당 URA로 전송한다. 그러면 이 요구를 받은 URA는 데이터베이스를 처리할 수 있는 URA로서 host에 user, pwd라는 아이디와 암호로 로그인하여, pub이라는 데이터베이스에 있는 table의 저장고에서 attribute = 'j'라는 판별 기준으로 원하는 데이터를 추출하여 매칭 에이전트에게 돌려준다.

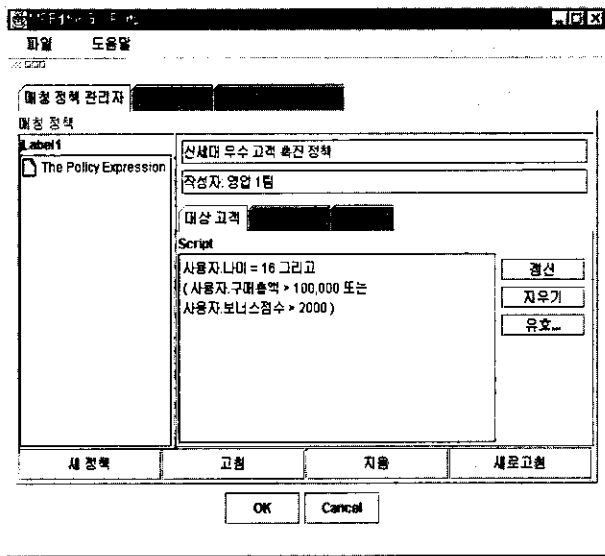
이때 주의할 것은 각각의 프로토콜마다 질의를 받는 방식이 다를 수 있지만 본 논문에서는 호스트, 컨테이너, 자료, 판별기준 등 네 가지를 구성요소로 하고, 구조는 URA의 문법을 참조하여 구현했다.

#### 4.3.5 정책 관리 톨(Matching Policy Editor)

정책 관리 톨은 정책관리자가 정책을 수립하고, 실시할 수

있도록 정책지향 에이전트와 통신하는 자바 클라이언트 프로그램이다. 자바 언어로 구성되어 이후 필요한 경우에는 모듈 별로 웹 페이지에서 동작하는 애플릿으로도 쉽게 변경될 수 있다. (그림 3)은 정책 관리 툴에서 정책을 편집하는 화면이다.

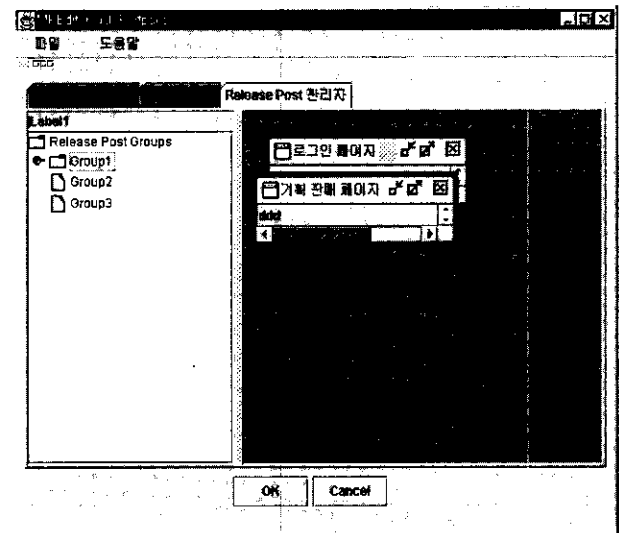
정책 관리 툴에서 정책관리자는 마케팅 정책을 간략한 스크립트를 이용하여 기술하게 된다. 이 기술언어는 자체 스크립트 엔진을 내장하고 있어 한글을 그대로 쓸 수 있다는 점이 장점이다. 그런 만큼 정책 관리자들이 보다 쉽게 정책을 관리할 수 있다. 하나의 정책을 수립하기 위해서 정책관리자는 대상고객 선정조건, 대상 아이템 선정조건을 입력하고 이들이 매칭될 때, 대응할 프로모션을 설정해야 한다. (그림 4)는 프로모션을 설정하는 화면이다.



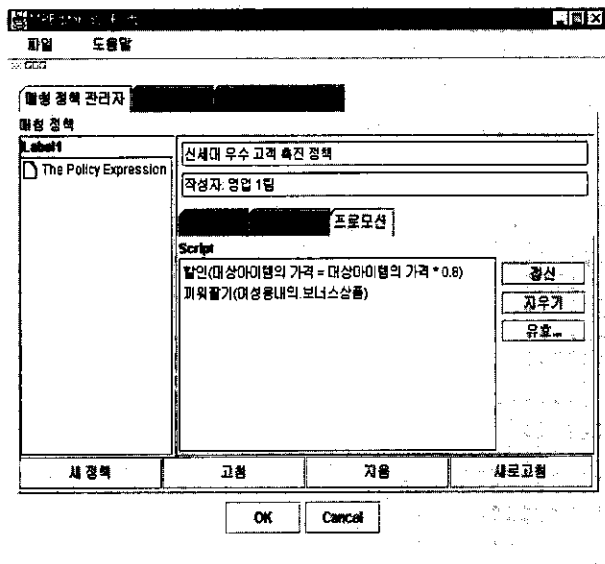
(그림 3) 정책 관리 툴에서 정책 정의 화면

위 그림에 나타난 프로모션 정책 기술 방식은 아직 충분히 다듬어지지 않았지만 기존의 룰 기반 매칭 시스템이 제공하는 인터페이스보다 자연어에 가까워 훨씬 이해하기 쉽고, 확장성도 있다. 다음으로 정책 관리 툴의 주요기능은 정책을 실시하는 것인데, 이 인터페이스를 나타낸 것이 (그림 5)이다.

정책 관리 툴을 통해 관리자는 정책이 실시될 Release Post를 볼 수 있고, 이들을 관리(생성, 편집, 삭제)할 수 있다. 위의 그림에도 나타난 바와 같이 Release Post는 동일한 사이트 내에 있는 서로 다른 웹 페이지일수도 있고, 또는 전혀 다른 사이트에 내보내진 광고가 될 수도 있다. 정책 관리자는 이들 Release Post에 앞에서 작성한 정책을 할당하여 실시할 수 있는 것이다.



(그림 5) 정책 관리 툴에서 정책 실시 화면



(그림 4) 정책 관리 툴에서 프로모션 설정 화면

## 5. 비교 및 고찰

### 5.1 사용 가능 예

본 연구에서 제안된 정책지향 매칭 에이전트는 아직 상용화되어 가동되고 있는 시스템은 아니고, 초기 버전인 시제품 성격의 솔루션이므로 사용가능 예를 중심으로 비교를 하겠다. 정책지향 매칭 에이전트는 기존의 룰 기반 매칭 에이전트를 포함하는 개념이므로 룰 기반 매칭 에이전트의 장점은 모두 가지고 있다고 할 수 있다. 기존의 룰 기반 매칭 에이전트의 단점은 다양한 프로모션의 구현이 어렵고, 정책의 실시에만 중점을 두었기 때문에 정책의 수립 및 평가가 어렵다는 점이다.

정책 관리자의 입장에서 룰 기반 매칭 에이전트와 정책지향 매칭 에이전트의 효율을 비교해 보자. 우선 프로모션의 다양성 면을 살펴보자. 기존 룰 기반 정책 에이전트가 아이템의 추천에만 강조를 하였던 반면 정책지향 매칭 에이전트

에서는 정책을 대상 고객, 대상 아이템, 프로모션으로 나누었기 때문에 추천 외에도 끼워 팔기, 묶어 팔기 등의 다양한 정책을 구현할 수 있다. 정책 관리의 일관성 면에서는 다음과 같다. 톨 기반 매칭의 경우 정책 관리자가 실시 톨을 만들어 놓으면 실시 공간에서 이벤트가 발생할 때마다 톨 리스트를 확인하고 해당 톨을 실행한다. 그렇기 때문에 정책관리자는 정책의 실시 공간을 구분하거나 지정할 수 없고, 정책의 평가를 웹 traffic을 분석하는 웹 마이닝이나 기본 컴포넌트의 교체에 의존해야하기 때문에 개별 정책이나 실시 공간별 평가가 어렵다. 반면에 정책지향 매칭 에이전트의 경우 정책관리자가 톨, 대상 고객, 대상 아이템이 결합된 정책을 공간별로 할당 할 수 있기 때문에 다양한 정책을 실시하거나 각 정책별 평가가 쉽다. 4.2절의 (그림 2)에서 기술한 Data Gatherer가 웹 traffic정보도 보내주기 때문에 정책의 수립도 쉬워진다.

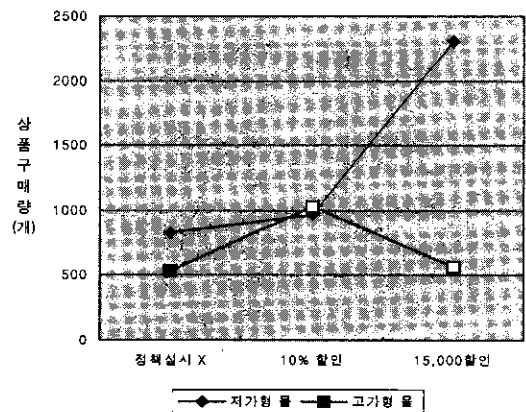
정책지향 매칭 에이전트에서는 정책을 시간에 대해서 종적으로 또는 횡적으로 실시, 분석할 수 있다. 시간에 종적이라는 말은 시간보다는 장소에 따라 서로 다른 정책을 적용할 수 있다는 뜻이다. 예를 들어 신촌 극장 앞과 명동 극장 앞에 각각 한 대씩의 kiosk가 있다고 가정했을 때, 동일한 검색엔진 이더라도 다른 kiosk에서 서비스되고 있다면 각 검색엔진의 가중치는 다르고 다른 검색엔진 이더라도 같은 kiosk에서 제공되고 있다면 가중치는 같다고 볼 수 있다. 이 경우 두 곳의 각 검색엔진마다 다른 정책을 한 달간 실시해서 각 장소의 kiosk에 가장 효과적인 정책을 찾아낼 수 있다. 또 다른 예는 한 곳의 kiosk에 두 가지 이상의 정책이 한달 동안 같은 확률로 임의로 실시되게 함으로써 그 곳에서 가장 효과적인 정책을 쉽게 찾을 수 있다. 그리고 시간에 횡적이라는 말은 장소보다는 시간에 따라 다른 정책을 적용할 수 있다는 의미이다. 예를 들어 신촌의 kiosk에서 시간대별로 둘 이상의 정책이 임의로 선택해서 실시되도록 함으로써 각 시간대별로 가장 알맞은 정책을 찾아낼 수 있다.

5.2 공간별 정책 구분 효과

5.1절에서 본 시스템의 장점의 하나로 실시 공간별로 관리자가 다른 정책을 실시할 수 있다는 점을 들었다. 본 논문에서는 이 점이 어느 정도의 효과를 가지는지 알아보기 위해 인터넷 상에서 설문조사를 하였다. 가상의 쇼핑 물 2개를 만들어 정책의 실시에 따른 구매 추이의 변화를 살펴보는 방식을 선택하였다. 여기서 실시 공간은 저가형 쇼핑 물과 고가형 쇼핑 물이 해당되고 정책은 '10% 할인'과 '15,000원 할인'으로 정하였다. 저가형 쇼핑 물에서는 10만원 미만의 상품만을 판매하며 고가형 쇼핑 물에서는 20만원대의 상품만을 판매하는 공간으로 설정하였다. 두 쇼핑 물은 모두 테니스 라켓, 소형 카세트, 전자 수첩, 카메라, 조명 기구, 무선 전화기, 의자, 고급 펜, 진공 청소기를 하나씩 판매한다. 설문

조사 방법은 기존 판매량을 알기 위해 첫 페이지에서는 아무런 정책을 적용하지 않고, 두 번째 페이지에서는 '10% 할인' 정책을, 세 번째 페이지에서는 '15,000원 할인' 정책을 실시하였다. 인터넷상에서는 다른 쇼핑 물로 빠르고 간편하게 이동할 수 있는 특성을 반영하기 위해 두 가상 쇼핑 물은 같은 페이지에 나타내었다. 설문 조사에 응한 사람은 총 531명이고 정책별, 물 별 총구매량은 (그림 6)과 같다.

(그림 6)을 보면 각 정책이 물에 따라 다른 효과를 가진다는 것을 알 수 있다. 10% 할인정책의 경우에는 고가형 물의 판매량이 저가형 물에 비해 81.4%가 더 증가했고, 15,000원 할인정책의 경우에는 저가형 물의 판매량이 고가형 물에 비해 175.9%가 더 증가했다. 결과가 이렇게 나온 이유는 각 정책별로 할인되는 정도가 다르기 때문인 것을 짐작할 수 있다. 예를 들어 테니스 라켓의 경우 저가상품은 24,000원이고 고가상품은 270,000만원이므로 10% 할인 정책을 실시하면 할인액은 각각 2,400원과 27,000원으로 고가상품은 저가상품의 11.25배가 할인된다. 그리고 15,000원 할인 정책을 실시하면 저가상품은 10% 할인되었을 때 보다 12,600원이 더 할인되는 반면, 고가상품은 10% 할인되었을 때 보다 12,000원이 더 비싸진다. 이처럼 정책이 실시 공간에 따라서 효과가 크게 다를 뿐만 아니라 매출액에 큰 영향을 미칠 수 있음을 확인했다.



(그림 6) 설문 조사 결과

6. 결 론

본 논문에서는 기존의 톨 기반 매칭에서 진화된 정책지향 매칭 에이전트 시스템을 제안하였다. 정책지향 솔루션은 마케팅 정책이 수립, 실시, 분석되는 각 과정에서 정책을 객체화하여 일관되게 관리할 수 있는 솔루션이다. 정책지향 솔루션에서 정책은 사용자와 아이템의 대상을 설정하고 여기에 프로모션을 결합하는 형태로 정의할 수 있다. 이와 같은 정의는 기존의 톨 기반 매칭 에이전트가 가지는 단점을 보완할 수 있다. 톨 기반 매칭 에이전트는 '개별화 아이템 선택'

기능에 머물고 있지만, 정책지향 매칭은 정해진 사용자와 아이템에 대해서 다양한 프로모션을 결합하는 형태를 취하기 때문에, 더욱 효과적인 마케팅 솔루션이라 할 수 있다. 또, 정책을 객체화했기 때문에 정책을 일관되게 추적할 수 있으며, 따라서 분석 활동도 다른 개념의 솔루션보다 고급화된 정보로 지원할 수 있다.

제안된 정책지향 매칭 에이전트 서버는 C++ 언어를 이용하여 window NT 상에서 구현하였고 클라이언트쪽의 정책 관리 틀은 java 언어를 이용하여 구현하였다. 구현 시 고려된 사항은 전자상거래를 위한 기초 플랫폼에 적합한 컴포넌트 기반 구조를 채택하였고, 기존의 여타 시스템에 연동되어 마케팅 활동을 쉽게 지원할 수 있도록 하였다. 또한 마케팅 활동을 중심에 둔 정책지향 솔루션이 되도록 하였고, 정책 담당자가 정책을 직관적으로 편집할 수 있는 인터페이스를 갖도록 하였다.

앞으로 더 연구되어야 할 것은 다양한 비즈니스 룰을 잘 표현할 수 있는 정책 언어를 개발하는 것이다. 본 연구를 통해서 자체 스크립트 엔진으로 정책을 기술하도록 하여 한글이 지원되는 등의 성과를 거두었지만, 이후에 보다 실제적인 비즈니스 환경에 대한 분석을 통해서 우리 실정에 맞는 비즈니스 룰 기술 문법을 개발할 수 있을 것이다.

### 참 고 문 헌

[1] Oracle, Oracle iMarketing Release 3i, <http://www.oracle.com>.

[2] P. Resnic, N. Iacocou, M. Sushak, P. Bergstrom, and J. Riedl, "GroupLens : An Open Architecture for Collaborative Filtering of Netnews," Proceedings of the Computer Supported Collaborative Work Conference, 1994.

[3] M. Ma, "Agents in E-Commerce," Communications of the ACM, Vol.42, No.3, pp.79-80, March, 1999.

[4] 황병연, 박성철, "전자상거래를 위한 정책지향 매칭 에이전트 시스템", 한국정보처리학회 13회 춘계학술대회발표논문집, 2000.

[5] P. Maes, R. H. Guttman, and A. G. Moukas, "Agents That Buy and Sell," Communications of the ACM, Vol.42, No.3, pp.81-91, March, 1999.

[6] J. B. Schafer, J. Konstan, and J. Riedl, "Recommender Systems in E-Commerce," Proceedings of the ACM Conference on Electronic Commerce, November 3-5, 1999.

[7] Global Home of Electronic Marketing, <http://www.ghem.com>, 1996.

[8] Broadvision, Building Dynamic Web Applications with Broadvision ONE-TO-ONETM, White Paper, <http://www.broadvision.com/>, April, 1997.

[9] A. Moukas, R. Guttman, and P. Maes, "Agent-Mediated Electronic Commerce : An MIT Media Laboratory Perspective," Proceedings of Int. Conf. On Electronic Commerce, Seoul Korea, pp.9-15, 1998.

[10] H. R. Rao, A. F. Salam, and B. DosSantos, "Marketing and the Internet," Communications of the ACM, Vol.41, No.3, pp.32-34, March, 1998.

[11] S. Alexander, "Web Marketing Gets Personal," InfoWorld Electric, 1/12/1997.

[12] FOLDOC(Free Online Dictionary of Computing), <http://foldoc.doc.ic.ac.uk/>.

[13] R. Kalakota and A. B. Whinston, 'Reading in Electronic Commerce', Addison Wesley Publishing Company, 1997.



### 황 병 연

e-mail : byhwang@songsim.cuk.ac.kr

1986년 서울대학교 컴퓨터공학과(공학사)

1989년 한국과학기술원 전산학과(공학석사)

1994년 한국과학기술원 전산학과(공학박사)

1999년~2000년 Univ. of Minnesota Visiting Scholar

1994년~현재 가톨릭대학교 컴퓨터전자공학부 부교수

관심분야 : 공간 데이터베이스(GIS), XML, WWW 데이터베이스, 전자상거래 등



### 박 성 철

e-mail : schpark@hotmail.com

2001년 가톨릭대학교 컴퓨터공학전공

(공학사)

관심분야 : 전자상거래, WWW 데이터베이스, XML, Learning Machine 등