

전자상거래 소비자 불만 접수용 에이전트 설계

윤영미^{*} 이미란 최영인 김연정 조동섭
이화여자대학교 과학기술대학원 컴퓨터학과
{little0, nayamira, cyi1109, chuwoo, dscho}@ewha.ac.kr

Agent Design for Reporting Consumer's Complaint Acceptance in E-commerce System

Young-mi Yoon, Mi-ran Lee, Young-in Choi, Yeun-jung Kim, Dong-sub Cho
Dept. of Computer Science & Engineering, Ewha Womans University

요약

최근 인터넷 사용자의 증가와 함께 전자상거래의 수요 또한 늘어나고 있다. 사용자는 인터넷 쇼핑몰을 이용하기 때문에 물건 구매 후 불만 사항이 있을 수 있으며 이는 e-mail을 통하여 접수될 것이다. 기존의 메일 기반 에이전트는 소비자가 e-mail로 접수한 소비자의 불만 사항들을 관리자가 하나하나 읽어보아야 하는 비효율적인 시스템이다. 관리자는 불만 처리에 대한 담당자가 아님에도 서버에 적재된 모든 메일을 읽어야만 하기 때문이다. 본 논문에서는 이러한 단점을 보완하기 위하여 도착한 e-mail을 자동으로 분류, 전달하는 시스템을 제안한다. 서버에 도착한 메일의 내용을 분석하여 해당하는 데이터베이스의 테이블에 저장하고 그에 대한 확인 메일을 소비자에게 발송하도록 하는 소비자 불만 접수용 에이전트를 설계하였다.

1. 서론

최근 인터넷의 대중화와 더불어 대표적인 산업으로 전자상거래 쇼핑몰에 대한 관심이 집중되고 있고 그에 따른 전자상거래의 중요성이 부각되고 있다. 인터넷을 기반으로 하는 전자상거래는 시간과 공간의 제약을 받지 않고 상품 판매가 가능하고 다양한 사용자 인터페이스를 제공함으로써 편리한 구매 및 판매 수단을 제공한다. 이러한 편의성 때문에 전자상거래를 이용하는 소비자의 수는 현저히 증가했지만 현재까지도 소비자의 욕구를 100% 만족시키지 못하고 있는 실정이다[1].

소비자는 자신의 상거래에 관한 불만 사항을 전자상거래 쇼핑몰에 e-mail을 통하여 보내고, 보내진 메일은 쇼핑몰 관리자에 의하여 처리된다. 이 때 서버에 도착한 메일을 회수하기 위하여 관리자는 메일 서버에 접속해야 한다. 메일 프로그램을 사용하여 자신의 PC로 가져온 후 확인해야 하고 또 메일의 내용을 일일이 읽어서 직접 분류하여 담당 부서에게

전달해야 하는 번거로움이 있다.

본 논문에서는 이러한 불편함을 줄이기 위하여 전자상거래 소비자 불만 접수용 에이전트 모델을 제안하고자 한다.

본 논문은 다음과 같이 구성되어 있다. 2절에서는 메일 시스템에 대한 기존의 연구에 대해 언급한다. 3절에서는 제안하는 전자상거래 소비자 불만 접수용 에이전트 모델의 전체 구성에 대해서 설명을 하고, 4절에서는 본 논문의 결론을 맺고, 향후 연구 계획에 대해 언급한다.

2. 관련 연구

2.1 전자우편 시스템

전자우편 시스템은 전자우편 클라이언트 프로그램으로 SMTP(Simple Mail Transfer Protocol)서버에 편지를 보낸다. 전자우편 메시지 수신인의 도메인 서버가 있는 도메인과 같으면 아스키 텍스트를 서버에 저장하고, 자기 도메인이 아닌 경우에는 해당 서버에 전달한다. 전자우편 메시지가 수신자의 SMTP

^{*} 이 논문은 2002년도 두뇌한국21사업에 의하여 지원되었음.

서버에 도착하면 수신자에게 할당된 사서함에 저장된다. 수신자는 아웃룩 익스프레스 같은 우편 클라이언트에서 POP3(Post Office Protocol) 프로토콜로 서버에 접속해서 자신에게 온 편지를 가져간다. 이 과정에서 포트 25를 사용하는 SMTP로 편지를 서버에게 전달하고, 포트 110을 사용하는 POP3 프로토콜로 편지를 클라이언트까지 배달한다.

일부 시스템(주로 유닉스 플랫폼)에서는 SMTP 편지 저장 공간에 직접 접근할 수 있다. 즉, 사용자가 SMTP서버에 접속해서 SMTP의 사서함으로부터 전자우편을 직접 읽을 수 있는 것이다. 그러나, 사용자의 관점에서 보면 아웃룩 익스프레스처럼 일단 사서함에서 클라이언트의 우편함으로 편지를 옮긴 후 읽는 편이 더 편리하다.

2.2 웹 기반 전자우편 시스템

웹 기반 전자우편 시스템은 계정과 서비스를 제공하는 서버와 사용자와의 인터페이스 역할을 담당한다. 브라우저를 통해서 전자우편 서비스를 제공하는 웹 사이트에 접속한 사용자는 ID와 암호를 가지고 자신의 정보를 관리하게 된다. 또한 사용자는 서버에 자신의 전자우편 주소를 가지게 되며 이를 이용하여 다른 사용자와 전자우편을 주고받을 수 있다. 전자우편 클라이언트 역할을 하는 브라우저는 사용자로부터 데이터를 입력받으며, 이는 HTTP(HyperText Transmission Protocol) 프로토콜을 이용하여 서버에 존재하는 CGI(Common Gateway Interface) 프로그램에게 전달된다. CGI 프로그램은 전달받은 사용자 데이터를 수정하여 실제로 전달 가능한 전자우편의 형태로 만들어 SMTP 프로토콜을 이용하여 수신자에게 전달된다.

이처럼 웹 기반 전자우편 시스템은 사용자가 특정 회사의 전자우편 클라이언트 프로그램을 구입하지 않아도 브라우저만 있는 환경이면 전자우편을 송수신할 수 있도록 함으로써 사용자에게 매우 편리한 환경을 제공한다. 이러한 환경은 기업이나 학교 등의 인트라넷 환경에 적용될 수 있으며 공동의 사용자 인터페이스를 제공하므로 업무 수행 능력이나 소속감 등을 높이는 데에도 기여할 수 있다[2].

웹 기반 전자우편 시스템을 통하여 수신된 메일들은 다양한 내용을 포함한다. 기업에서 인트라넷 환경에 웹 기반 전자우편 시스템을 적용하여 사용할 경우 기업은 매우 많은 양의 메일을 수신할 것이고 그 내용 또한 방대할 것이다. 사용자가 보내는 방대

한 양의 메일을 데이터베이스에 저장하기만 한다면 관리자의 입장에서는 메일들을 일일이 읽어보아야만 하는 번거로운 작업이 된다. 따라서 수신된 메일을 적절하게 분류하여 목적에 맞는 부서로 메일을 자동으로 보내주는 에이전트가 필요하다.

3. 불만 접수용 에이전트의 설계

기존의 메일 기반 시스템은 사용자가 웹사이트를 통하여 입력한 불만 사항이 판매 회사의 불만 접수 데이터베이스에 무분별하게 모두 저장되는 시스템이다. 이는 접수된 모든 글들을 관리자가 일일이 읽어보아야 하는 비효율적인 시스템이다. 서버에 메일을 적재하기만 하는 방식이기 때문에 관리자가 직접 모든 메일의 내용을 읽어보고 불만 사항을 분류하여 해당 부서로 다시 보내야 하는 번거로움이 있다. 따라서 사용자와 관리자의 개입 없이 번거로운 작업들을 대행해 주는 시스템이 필요하다.

소비자 불만 접수용 에이전트는 서버에 적재되어 있는 메일을 POP3 서비스를 이용하여 관리자의 PC로 가져오고 각 메일의 본문을 특정 단어(Keyword)의 유무를 기준으로 검색한다. 분류를 통하여 메일의 카테고리를 결정한 후에는 이미 구성되어 있는 메일 데이터베이스의 각 카테고리 테이블에 메일을 저장한다. 사용자는 자신의 불만 사항을 전자우편으로 관리자에게 보내기만 하면 에이전트를 통해 자동으로 분류가 되고 메일 데이터베이스의 각 테이블에 저장되게 된다. 저장된 메일은 각각의 담당 부서로 보내지고 불만 접수 처리에 대한 확인 메일 또한 자동으로 소비자에게 보내지도록 한다. 이는 관리자의 수작업이 필요 없는 시스템이므로 기존의 메일 기반 에이전트 시스템에 비하여 시간과 노력을 절감할 것을 기대할 수 있다.

3.1 전체 구조

사용자는 인터넷 쇼핑물을 이용하기 때문에 물건 구매 후 불만 사항이 있을 수 있으며 이는 e-mail을 통하여 접수될 것이다. 기존의 메일 기반 에이전트는 소비자가 e-mail로 접수한 소비자의 불만 사항들을 관리자가 하나하나 읽어보아야 하는 비효율적인 시스템이다. 관리자는 불만 처리에 대한 담당자가 아님에도 서버에 적재된 모든 메일을 읽어보아야만 하기 때문이다. 이러한 단점을 보완하기 위하여 수신된 e-mail을 자동으로 분류, 전달하는 시스템을 제안하

고자 한다.

서버에 도착한 메일의 내용을 분석하여 해당하는 데이터베이스의 매칭되는 테이블에 저장한다. 저장된 내용을 해당 부서의 담당자에게 보낸 후, 그에 대한 확인 메일을 사용자에게 발송하도록 하는 소비자 불만 접수용 에이전트를 설계하였다.

본 논문에서 제안하는 불만 접수용 에이전트의 전체 구성은 다음 그림 1과 같다.

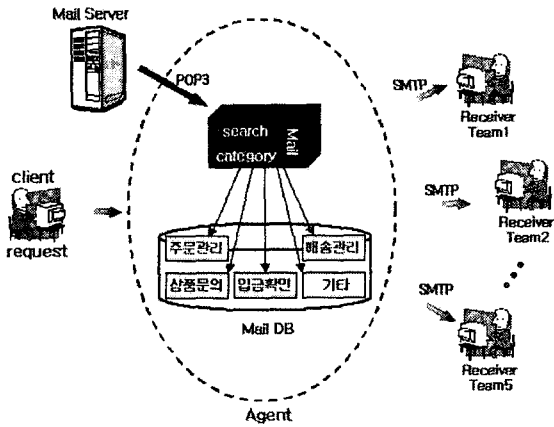


그림 1. 에이전트의 전체 구성

각 과정은 다음과 같은 단계를 거친다.

- ① 사용자는 자신의 불만 사항을 e-mail로 작성하여 관리자에게 보낸다.
- ② 에이전트는 시간대별로 메일을 읽어 특정 단어 (Keyword)를 추출하여 간단한 분류 과정을 거친다.
- ③ 분류 과정을 거친 메일은 메일 데이터베이스 내의 매칭되는 테이블에 저장되고 결과 메시지를 에이전트에 돌려준다.
- ④ 테이블에 저장되어 있는 메일은 에이전트에 의해 해당 부서의 담당자에게로 보내진다.
- ⑤ 에이전트는 사용자에게 불만 처리 확인 메일을 보냄으로 불만 접수 처리를 마친다.

본 연구에서는 데이터베이스에의 저장을 위한 필드들을 새롭게 정의하여 사용하고 있다. 특정 단어 (Keyword) 검색 방법을 이용하기 때문에 매칭 테이블을 결정하기 위한 키워드를 지정해 주어야 한다. 표 1은 테이블 분류를 위한 키워드를 나타낸다.

에이전트를 어떻게 설계하느냐에 따라 키워드의 내용이 달라진다. 키워드의 예로는 주문, 언제, 은행

등이다.

표 1 테이블 분류를 위한 키워드

테이블 이름	필드
주문관리(order)	주문,개(수량),취소,환불,영수증
배송관리(send)	배송,도착,언제,시간
상품문의(goods)	크기,색,구입
입금확인(receipts)	입금,은행,카드,계좌,예금주
기타(etc)	제시한 키워드 외

키워드 추출 시 Loop를 이용하여 본문 text를 검색하도록 한다.

이러한 분류 과정을 거쳐서 수신된 메일은 메일 데이터베이스의 매칭되는 테이블에 저장된다. 에이전트는 테이블에 저장된 메일들을 해당 부서의 담당자에게 보내고 사용자에게 SMTP 서비스를 이용하여 처리 확인 답장을 보내게 된다.

3.2 메일 전송 시스템

사용자가 보낸 불만 접수 메일들은 한 곳의 메일 서버에 도착하게 된다. 메일 서버에 저장된 메일들은 메일 내용의 키워드 검색을 통하여 분류 과정을 거치게 되는데 이에 따라 데이터베이스 내의 저장될 매칭되는 테이블들이 결정된다. 이 때 일정한 시간이 경과하면 서버에 새롭게 수신된 메일이 있는지를 확인한 후 새로운 메일이 있을 경우 메일 데이터베이스에 저장하도록 한다. 본 논문에서는 보안 처리 및 관리자의 편리성을 위하여 하나의 데이터베이스에 모든 정보를 포함시키도록 하였으며 테이블은 order table, send table, goods table, receipts table, etc table로 이루어져 있다.

수신된 메일은 name, email, date, title, contents의 필드별로 파싱된다. 에이전트는 테이블에 저장된 메일들을 해당 부서의 담당자에게 보내고 사용자에게 SMTP 서비스를 이용하여 처리 확인 답장을 보내게 된다.

그림 2는 POP으로 가져온 새로운 메일의 수 (INumMessages)가 0보다 클 때 실행되는 것으로 VC++로 표현한 것이다. 이것을 살펴보면 먼저 메일을 텍스트 파일로 변환하고 변환된 파일에서 각각의 필드에 넣을 데이터를 찾아 변수에 저장한다. 그리고 난 후 본문의 내용에서 키워드를 찾아 check값을 결정한다. 데이터를 각각의 테이블에 저장하기 위해 switch문을 사용하여 쿼리를 결정한 뒤 쿼리를 실행한다.

```

//새로운 메일의 수만큼 실행
for (int iMessageId=0; iMessageId<(int) INumMessages; iMessageId++)
{
    char szFileName[MAX_PATH] = "";
    int ch=0;
    // mail을 txt파일로 변환
    sprintf(szFileName, "%s-%s-%04d.txt", pszServer,
        pszUserName, iMessageId + 1);
    ...
    // 각각의 필드에 넣을 데이터를 찾고 각각의 변수에 저장한다.
    ifstream f3(szFileName);
    while(!f3.eof()){
        if(!f3) AfxMessageBox("error");
        f3>>n;
        // name 필드에 넣을 데이터를 찾는다
        if(strcmp(n,"From:")=0){
            f3>>n;
            break;
        }
    }
    f3.close();
    ...
    // keyword에 따라 check값 결정
    if(strstr(temp_c,"주문")|| strstr(temp_c,"개")
        || strstr(temp_c,"환불"))
        check='o';
    ...
    // keyword에 따라 달라진 check값에 따른 쿼리문
    // n: name , e: e-mail , d: date , t: title , c: contents
    switch(check){
        case 'o':
            SqlString2.Format("INSERT INTO orders VALUES
                ('%s','%s','%s','%s','%s')",n,e,d,t,c);
            break;
        ...
        case 'r':
            SqlString2.Format("INSERT INTO receipts VALUES
                ('%s','%s','%s','%s','%s')",n,e,d,t,c);
            break;
        default:
            SqlString2.Format("INSERT INTO tot VALUES
                ('%s','%s','%s','%s','%s')",n,e,d,t,c);
    }
    // 쿼리문 실행.
    database.ExecuteSQL(SqlString2);
    ...
}

```

그림 2. 수신된 메일의 분류 및 저장 알고리즘

4. 결론 및 향후 연구과제

본 논문에서 제안하는 소비자 불만 접수용 에이전트는 POP Mail Service와 Content Transfer

Encoding 8bit를 지원하는 서버에 한하여 에이전트 기능을 수행하도록 한다. 소비자 불만 접수용 에이전트는 사용자가 보낸 모든 메일을 관리자의 데이터베이스에 저장하게 되므로 이전 데이터에 대한 검색이 매우 용이하다. 기존 사용자가 주로 어떠한 불만 사항을 접수했는지에 관한 데이터를 쉽게 얻을 수 있기 때문에 과거 데이터의 통계 결과를 얻을 수 있고 그것을 토대로 문제점 및 보완점을 찾아내어 전자상거래 시스템 설계에 적용할 수 있다는 장점이 있다.

향후 연구과제는 제안된 소비자 불만 접수용 에이전트를 실제 전자상거래 사이트에 적용하여 소비자의 불만을 실시간으로 처리하는 시스템을 구축하는데 있다. 그리고 성능 향상을 위하여 자연어 검색을 지원하는 에이전트를 구축하여 메일 에이전트의 자연어 처리 문제, 에이전트에서의 지식 베이스 구축 문제와 같은 시스템 확장 문제 등을 연구할 것이다.

참고문헌

- [1] 김명순, 정환목, "자연어 질의가 가능한 퍼지 기반 지능형 전자상거래 검색 에이전트," KFIS Spring Conference, 2001. 5.
- [2] 임양원, 권기훈, 임한규, "서비스 엔진을 이용한 웹 기반 메일 에이전트 시스템의 설계 및 구현," 한국정보처리학회 논문지 제7권 제2호, 2002. 2.
- [3] 이해수, 장덕성, "JATLite 기반 멀티 에이전트 시스템의 설계 및 구현," 한국정보처리학회 논문지 제7권 제8호, 2000. 8.
- [4] Leon Braginski and Matt Powell, "Running Microsoft Internet Information server," Microsoft Press, 1998.
- [5] David G.Schwartz, "When email meets organizational memories: Addressing threats to Communication in a learning organization," International Journal of Human-Computer Studies, 51, Article No. ijhc.1999.0276, pp.599-614, 1999.