韓國 컴퓨터情報學會 論文誌
第11卷 第4號, 2006. 9.

2006-11-4-4-2

닷큐어리를 활용한 대화형 검색 에이전트

김 선 옥 *

An Interactive Search Agent based on DotQuery

Kim SunOk *

요 약

인터넷의 발달로 웹상에 수많은 문서와 웹서비스가 급격히 증가하고 있다. 인터넷 사용자들은 웹에서 원하는 정보와 서비스를 획득하기 위해서는 각종 브라우저에서 여러 단계의 프로그램 조작 절차를 반복하고 사이트마다 고유의 서비스 구조들을 이해한 상태에서 여러 번의 방문 절차들을 거쳐야 한다. 이러한 절차들은 사용자에게 꼭 필요한 선행 작업이며, 사용자가 실제로 서비스 획득에 걸리는 시간보다도 선행 작업에 소요되는 시간이 더 큰 게 현실이다. 이러한 시간소요를 극복하기 위해 단순 반복적인 작업을 보다 체계화하고 단순화하기 위한 방안으로 닷큐어리 기반의 대화형 검색 에이전트를 제안한다. 제안하는 에이전트는 사용자의 컴퓨터에서 자연어 명령을 포함하는 닷큐어리를 통해 인터넷 사이트가 제공하는 다양한 서비스를 획득하는데 소요되는 여러 절차들을 대행하게 하고, 다수의 웹사이트에 대한 병렬 서비스들 또한 대행하게 한다. 이 에이전트는 IE와 같은 범용 브라우저 내에 플러그 인되어 닷큐어리를 해독하며, 자체 프로그램을 통하여서도 사용자가 지시한 닷큐어리를 분석하여, 복수의 자체브라우저들을 통해 서비스 결과를 획득하도록 구성하였다.

Abstract

Due to the development of Internet, number of online documents and the amount of web services are increasing dramatically. However, there are several procedures required, before you actually find what you were looking for. These procedures are necessary to Internet users, but it takes time to search. As a method to systematize and simplify this repetitive job, this paper suggests a DotQuery based interactive search agent. This agent enables a user to search, from his computer, a plenty of information through the DotQuery service, which includes natural languages, and it executes several procedures required instead. This agent also functions as a plug-in service within general web browsers such as Internet Explorer and decodes the DotQuery service. Then it analyzes the DotQuery from a user through its own program and acquires service results through multiple browsers of its own.

▸ Keyword : Web site, Interactive search, Natural language, Information search

---

• 제1저자 : 김선옥
• 접수일 : 2006.08.12, 심사일 : 2006.09.11, 심사완료일 : 2006.09.23
* 한라대학교 정보통신공학부 전임강사

Ⅰ. 서 론

기존의 온라인 및 오프라인 서비스들이 인터넷을 통한 서비스로 제공되고 있으며 기하급수적으로 증가하는 이러한 정보 및 서비스로 인하여 웹에서 원하는 정보와 서비스를 찾아 이용하는데 많은 시간과 노력이 소요되고 있다. 많은 프로그램들이 사용자가 필요한 정보들을 찾고 이용하는 것을 돕고 있지만, 효율적이고 편리한 방법을 제공하기에는 충분하지 않다. 이는 웹사이트에서 다루어지는 정보의 양과 종류가 증가함에 따라 구성이 복잡해졌기 때문이다. 이렇게 복잡하게 구성된 웹사이트에서 사용자가 신속하고 간단하게 원하는 서비스를 획득하기 위해서는 웹브라우저 프로그램의 반복되는 조작과 더불어 방문하고자 하는 웹사이트로의 이동, 원하는 서비스 웹페이지로의 이동, 서비스 획득에 필요한 절차(클릭, 키워드입력, 조건입력 등)들이 필요하다. 또한, 웹을 통해 원하는 정보를 얼마나 신속하고 정확하게 검색할 수 있는지를 나타내는 웹 접근 방식은 인터넷 환경에서 중요하다[1]. 하지만, 현재의 웹 접근 방식에는 다음과 같은 문제점을 가지고 있다.

첫째, 사용자들이 선호하는 이용 패턴 내에서 방문하는 웹사이트 및 제공 서비스들은 무수히 많으나 사용자들은 이러한 사이트 및 서비스로의 이동이 항상 동일하게 반복적인 검색과정으로 이루어진다.

둘째, 실질적으로 사용자가 원하는 인터넷 서비스를 수행하기 위해 웹사이트에 접속한 경우 사전 작업이 필요한 경우가 많다. 메일이나 전자상거래와 같은 인터넷 서비스를 제공하는 웹사이트의 경우에는 사용자 아이디 및 패스워드를 입력하는 작업이 항상 선행되어야 한다. 또한, 웹사이트마다 사용자ID를 다르게 등록한 경우에는 입력 실수 등이 발생하기 쉽다.

셋째, 사용자가 원하는 인터넷 서비스를 얻기 위해서는 해당 서비스의 웹사이트 주소를 사용자가 일일이 기억하여야 한다.

넷째, 일반적으로 인터넷 서비스 이용 시에는 문자열을 입력하여야 한다. 이때, 키보드를 이용하여 문자열을 입력하거나, 입력장치의 표시 자를 이용하여 블록을 설정하여 복사, 붙여넣기 등을 통하여 문자열을 입력하여야 하므로 부가적인 키 작업이 수반되어야 한다.

이러한 문제점들은 개별화되어 복잡하게 구성된 수 많은 정보제공 웹 사이트에서 서비스를 받기 위한 일련의 과정 속에서 표현되는 일상적인 작업이다. 따라서 본 논문에서는 사용자의 개인 컴퓨터를 마치 웹 서비스 제공 사이트처럼 가상화하여, 개인화 된 맞춤형 검색 서비스를 사용자 스스로 정의하며, 컴퓨터 사용 시에 단순 반복 작업을 단순화시키는 IE기반의 대화형 검색 에이전트를 제안한다. 이 에이전트는 닷큐어리 서비스, 스케줄 기능, 복수 브라우저 창을 편리하게 네비게이션 할 수 있는 전용 브라우저 등의 기능을 제공하는 동시에 다중 정보검색을 대행할 수 있다. 우리는 IE기반의 대화형 검색 에이전트 소프트웨어를 통해 개인화 된 맞춤형 검색 서비스를 사용자 스스로 설정하며, 정보검색을 할 때 사용자가 단 한 번의 클릭으로 원하는 다중의 사이트에서 동시에 정보를 획득할 수 있는 원-클릭 서비스를 제공 받을 수 있다. 그리고 병렬처리 웹 서비스 등의 기능을 제공받아 여러 종류의 다양한 웹 서비스들을 하나의 사용자 정의 웹 서비스로 등록하여 빠른 웹의 접근을 도모한다.

본 논문에서는 사용자와 대화를 통해 필요한 서비스를 제공해주는 고속검색의 대리자 역할을 수행하는 대화형 에이전트를 위해, 사용자 질의 분석을 통한 서비스 수행을 담당하는 시스템을 구축하여 검증하고자 한다. 본 논문의 구성은 다음과 같다. 먼저, 2장에서 일반 브라우저 검색방식에 대한 문제점을 제시하여 개선점 및 대화형 검색 에이전트 관련 연구의 필요성을 설명하며, 3장에서는 제안하는 대화형 검색 에이전트의 구조 및 구현에 대하여 설명한다. 4장에서는 대화형 검색 에이전트를 활용한 간단한 예시 및 결과를 제시하며, 5장에서는 결론으로서 제안하는 소프트웨어의 필요성 및 결과에 대하여 서술한다.

Ⅱ. 관련연구 및 개선방안

인터넷 서비스 분야에서 원하는 정보를 얻기 위한 빠르고 정확한 접근방법에 대한 많은 연구가 진행되어 왔다. 특히 WI(Web Intelligence) 분야에서는 웹 정보에 대한 검색, 웹 에이전트 등의 다양한 연구가 이루어지고 있다[2]. 이들 연구는 웹 정보를 표현하는 기존의 방법에서 벗어나 보다 빠른 웹의 접근에 관한 방법을 연구하고 있다.

2.1 사이트 접근 기법

Divyakant A.등은 정보의 단위를 관련된 웹 페이지들의 묶음으로 보고, 서로 관계된 정보가 있는 페이지들을 검색하여 웹의 정보에 접근한다[3]. 이 연구는 연관된 정보들을 함께 볼 수 있는 장점이 있다. Minqing H. 등은 페이지 내에 있는 정보를 검색하여 웹에 대한 접근을 시도한다

[4]. 이는 하나의 페이지에 여러 주제의 정보가 있지만, 질의어의 분포가 하나의 구역에 집중할수록 페이지에 포함된 질의어들이 같은 정보를 나타냄을 이용한 것이다.

하지만, 위의 연구들에서 알 수 있듯이 웹 페이지는 하나의 브라우징 단위로 단말기의 화면에 정보를 나타내기 때문에 빠른 정보를 얻기 위해서는 한계가 있다. 이러한 문제점들을 해결하기 위해 다음과 같은 대화형 검색 에이전트를 제안한다.

첫째, 하나의 웹 페이지는 여러 종류의 정보를 포함하고 있어 사용자의 의도와 연관성이 없는 정보들이 검색에 영향을 준다. 따라서 본 논문의 대화형 검색 에이전트에서는 관련된 정보들을 입력 창에 입력하여 복수 브라우저 창으로 탐색하여 연관성이 있는 정보들만 검색한다. 이렇게 함으로 페이지 내의 반복되는 정보와 광고 등의 불필요한 정보의 접근을 막아 탐색시간을 절약 할 수 있다.

둘째, 검색엔진의 결과는 검색된 각 페이지의 URL로 이동할 수 있는 링크를 제공한다. 하지만 웹사이트 내의 기본적인 구성은 대부분 첫 페이지 즉 웹 사이트의 메인 페이지로부터 방문을 시작한다는 가정을 바탕으로 작성되므로 원하는 검색결과를 얻기 위해서는 여러 웹 페이지로 이동이 필요하다. 하지만, 본 논문에서 제시한 대화형 검색 에이전트에서는 한 페이지로부터 시작되는 일련의 링크 페이지들을 검색하여 본 논문의 브라우저 에이전트 선택 창에 등록시켜 원하는 웹 페이지의 직접 접근을 수행할 수 있다.

2.2 재방문시의 접근 기법

Shaun K.는 "back"과 "history" 그리고 "bookmark" 기능을 통합해 사용자의 별도의 요청 없이도 자동으로 사용자가 자주 방문하는 페이지를 북마크 한다[5]. Teruhisa M. 등은 페이지 내의 관심 있는 문장에 북마크할 수 있도록 함으로써, 재접근 시에 해당 컨텐츠의 위치로 빠르게 이동할 수 있다[6]. 따라서 텍스트가 많은 페이지 내에서 정보의 재접근을 시도할 때 시간은 다소 단축할 수 있다. 그러나 이러한 연구방법은 저장된 북마크 페이지를 북마크 리스트 중에서 다시 찾아내야 하는 번거로움과 페이지 단위의 북마크로 인해 사용자의 의도를 정확하게 저장하지 못하는 단점이 있다. 본 논문에서는 브라우저 선택 창을 이용하여 이런 문제점들을 해결하고자 한다. 즉, 사용자가 이동한 여러 페이지들의 링크를 통한 경로를 브라우저 에이전트 선택 창에 등록시켜 재방문 시에 사용자가 단 한 번의 클릭으로 원하는 사이트에서 접근할 수 있는 원-클릭 서비스를 제공한다. 본 논문에서 제시한 브라우저 에이전트 선택 창은 사용자가 자주 접하는 정보를 동시에 여러 페이지에 접근할 수 있게 한다. 또한 여러 페이지에 흩어진 관련 정보를 함께 볼 수 있는 병렬처리 웹 페이지 등의 기능이 있어 불필요한 이동 검색 횟수를 줄일 수 있다. 또한, 여러 개의 웹 페이지들을 띄워 놓는 탭 브라우징 기능을 제공하며 하나의 사용자 정의 웹 페이지로 등록할 수 있게 한다.

III. 제안 검색 에이전트의 자연어 검색 처리 구조

본 논문에서 제안하는 대화형 검색 에이전트의 기능별 구성 모듈은 그림1 과 같은 구조로 이루어져 있다. 주요 구성모듈은 "닷큐어리 분석기"와 "사이트/서비스 메니저"와 "후킹 메니저" 그리고 "Job 스케줄러"로 구성된다. 사용자는 전용 브라우저 혹은 범용브라우저의 주소 창을 통하여 닷(.)으로 시작하는 질의를 입력하면 "닷큐어리 분석기"가 입력 질의를 분석하여 원하는 사이트, 원하는 서비스를 "사이트/서비스 메니저"를 통하여 최종의 타겟 URL을 생성하게 된다. 이렇게 생성된 URL은 내부와 외부의 브라우저를 통해 자동으로 해당 사이트로 방문하여 원하는 서비스를 수행하게 된다.

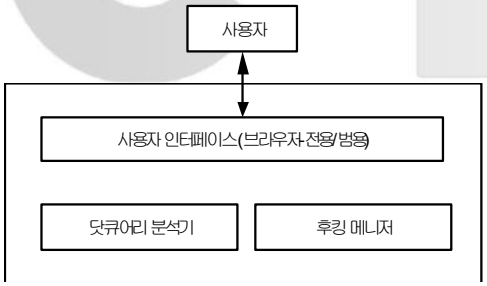

그림1. 대화형 검색 에이전트의 구성 모듈
Fig1. Functional modules of interactive search agent

대화형 검색 에이전트의 기능 요소 중 닷큐어리를 사용자로부터 입력받아 서비스 메니저를 통한 웹사이트 서비스를 수행하는 닷큐어리 분석기 및 사이트/서비스 메니저의 알고리즘의 구체적인 단계는 그림2와 같다.

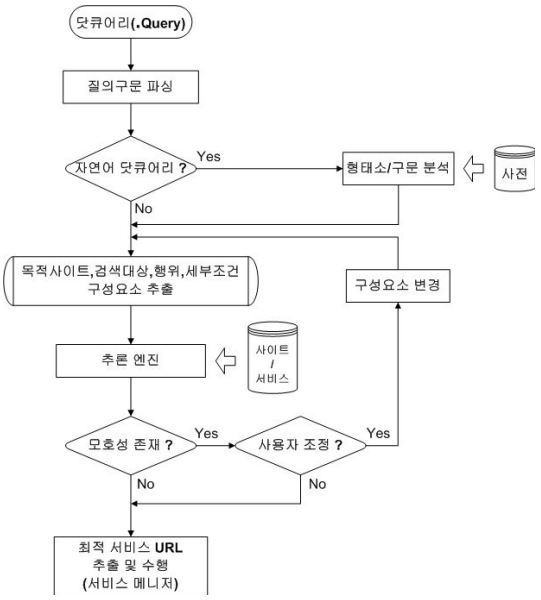

그림2. 닷큐어리 분석기 및 서비스 메니저의 알고리즘
Fig2. Algorithm of DotQuery and Service Manager

위의 알고리즘에서와 같이 사용자가 입력한 닷큐어리를 닷큐어리 분석기가 구문파싱하게 되고, 해당 질의가 자연어 기반 닷큐어리일 경우에는 형태소분석 및 구문분석 과정을 거쳐 목적사이트, 검색대상, 행위, 세부조건 구성요소로 추출된다. 만일 사용자 질의가 서브 닷큐어리인 경우에는 별도의 추출과정이 생략 된다. 이렇게 추출된 4개의 구성요소를 기반으로 추론엔진은 최적의 사이트 및 서비스 URL 을 추론하게 되는데, 사용자의 입력 실수 및 모호한 질의어일 경우에는 사용자의 확인 과정을 거쳐, 입력 질의를 재수정하거나 강제 수행을 진행할 수 있다. 이와 같이 사용자 닷큐어리 분석과정을 거쳐 도출된 4개의 구성요소는 추론엔진을 경유하여 최적의 서비스 URL이 생성되며, 생성된 URL이 1개 이상 일 경우에는 사이트/서비스 메니저가 여러 개의 내부 브라우저를 통해 병렬적인 서비스 실행을 담당하게 된다.

3.1 닷큐어리(.Query) 분석기

닷큐어리 분석기의 동작 형태는 그림3 와 같으며 사용자의 질의에 대한 정확한 의도를 추론하며 추론 된 정보를 가지고 "사이트/서비스" 메니저가 필요로 하는 그림4 와 같이 목적사이트, 검색대상, 행위, 세부조건 항목을 추출한다.

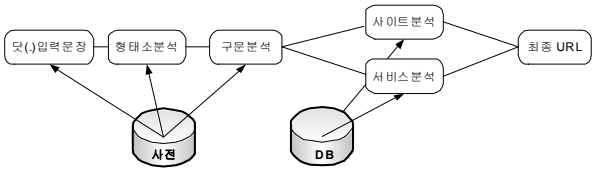

그림3. 닷큐어리 분석기의 동작 형태
Fig3. Movement Pattern of DotQuery Analysis

주소 창을 통해 입력된 질의어를 구분하면, 닷(.)이 없는 경우와 닷(.)으로 시작하는 경우(닷큐어리)의 두 종류로 구분한다.

첫째, 닷(.)이 없는 경우는 영문 도메인 또는 자국어 도메인일 경우이다. 이러한 예로 www.yahoo.co.kr와 청와대 같은 사이트 명이 있다. 이 경우에는 곧바로 해당 페이지에 접근한다.

둘째, 닷(.)으로 시작하는 경우로 사용자의 질의를 자연어 형태소 분석 과정을 거쳐서 그림4와 같이 목적사이트, 검색대상, 행위, 세부조건 항목들을 추출하는 과정이 필요하다("자연어 기반 닷큐어리"). 또 다른 방법으로는 각 항목 사이에 서브닷(.)을 인위적으로 입력하는 방법이다("서브 닷 기반 닷큐어리").

닷큐어리 유형은 크게 자연어를 기반으로 한 큐어리와 서브 닷을 통한 큐어리로 구분할 수 있는데, 각각의 유형별 장단점이 존재한다.

"자연어 기반 닷큐어리"는 사용자 의도를 정확하게 분석하여야 하며 만일 의도하지 않은 방향으로의 웹서비스 대행이 이루어질 경우 신뢰도에 큰 타격을 입게 된다. 자연어를 기반으로 하기 때문에 컴퓨터 미숙자들에게는 아주 편리한 기능을 제공할 수는 있지만 자연어 인식 율이 떨어질 경우에는 그 효용성이 반감되기도 한다. 또한 자연어 처리를 하기 위한 각종 형태소분석, 구문분석 등의 과정을 거치기 위해서는 매우 많은 컴퓨터 자원(사전, 통계, CPU 등)이 소모된다는 단점도 있다. 이러한 한계를 극복하기 위해서는 인식률이 높고 처리 속도가 빠른 자연어 처리 알고리즘이 동반되어야 하며 그에 대한 연구 및 고찰은 현재 꾸준히 지속되고 있다.

"서브 닷 기반 닷큐어리"는 "자연어 기반 닷큐어리"의 기술적 한계를 극복하기 위한 한 방법으로, 에이전트에게 좀 더 명쾌하고 정확한 명령을 지시할 수 있는 장점을 가진다. 즉 에이전트가 잘못 판단하여 의도하지 않은 결과를 도출하는 것을 방지할 수 있으며 컴퓨터 자원 소모를 최소

화할 수 있는 장점이 있는 반면 서브 닷큐어리의 규칙을 준수해야만 하는 단점이 존재한다. 즉 반드시 서브 닷큐어리 구성요소간의 구문규칙(목적사이트. 검색대상. 행위. 세부조건)을 지켜야만 한다. 물론 구문규칙의 구성요소 순서는 사용자 설정에 따라 수정 가능하도록 구성한다.

위의 두 가지 유형의 닷큐어리에서 추출된 구성 요소 중 "목적사이트", "행위" 요소에 대해서는 정형화된 데이터베이스 구축이 관건이며, 사용자의 잘못된 입력(오타)에 대비한 추론 로직 또한 부가되어져야 한다.

추론과정에서의 모호성을 해결하기 위해서는 사용자의 환경설정에 따라, 잘못된 추론일지라도 서비스대행을 강제할 수 있도록 한다거나, 서비스대행 이전에 추론과정에서 도출된 정보를 사용자에게 먼저 확인받도록 하는 중간단계를 경유할 수 있도록 구성한다.

본 논문에서는 두 가지 경우 중에서 서브 닷큐어리를 우선 구현하여 사용자의 신속하고 간편한 웹서비스 취득을 가능토록 한다.

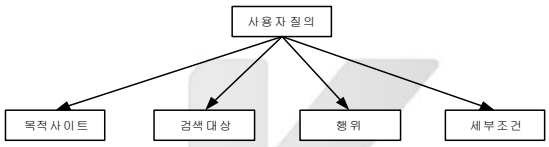

그림4. 목적사이트, 검색대상, 행위, 세부조건 항목 추출

Fig4. Target Site, Search keyword, Action, Extraction of detail conditions item

사용자 질의에 대하여 닷큐어리 분석기를 통한 추출 항목들은 크게 다음의 세 가지 유형으로 구분할 수 있다.

유형1: 어느 사이트에서 이루어지는가?
(예: . 네이버)

유형2: 어느 사이트로, 무엇에 대하여, 어떤 행위를 해야 하는가?
(예: .네이버에서 유비쿼터스에 대한 검색을 한다.)

유형3: 어느 사이트로, 무엇에 대하여, 어떤 행위를 단 세부 조건은 어떻게 이루어지나?
(예: .다나와에서 MP3에 대해 가격비교를 하지만 10만원 이내 것으로 한다.)

위의 세 가지 유형에서 4개의 항목(목적 사이트, 검색대상, 행위, 세부조건)을 추출하기 위해서는 자연어 처리 색인을 이용한 분석을 한다. 그러나 이러한 추출 과정에는 상당히 많은 색인 정보를 보유하고 있어야 하며 구축에 많은 비용과 시간이 소요된다. 본 논문에서는 이러한 자연어 처리의 한계 및 불완전한 인식률 극복을 위한 대안으로 그림5와 같은 문법체계를 이용하여 자연어 처리의 한계를 대체하기 위한 서브 닷큐어리를 제안한다. 이 방법은 사용자의 정확한 의도를 파악하는데 가장 경제적이면서도 가장 정확한 수단을 제공한다.

목적사이트 . 검색대상 . 행위 . 세부조건

그림5. 서브닷큐어리를 이용한 자연어처리 한계 대체

Fig5. Limitations Overcome of Natural Language Treatment using SubDotQuery

다음은 자연어 기반 서브 큐어리 이며 괄호 안의 내용은 사용자 의도를 정확히 나타내는 서브 닷큐어리의 예를 나타낸다.

예1: .네이버에서 특허자료 중 유비쿼터스에 대해 검색하라
(서브닷1:.네이버.유비쿼터스.검색.특허자료) 혹은
(서브닷2:.네이버.특허자료 중 유비쿼터스.검색)

예2: .야후에서 논문자료 중 유비쿼터스에 대해 검색하라.
(서브닷1:.야후.유비쿼터스.검색.논문자료) 혹은
(서브닷2:.야후.논문자료 중 유비쿼터스.검색)

목적사이트를 내용에 따라 나누면 그림6의 "목적사이트" 구성요소와 같이 크게 세 종류로 분류된다. 첫째, 사용자가 필요에 의해서 설정한 사이트일 경우에는 "나의", 혹은 "내가" 등의 자아에 대한 주어가 올 수 있다. 둘째, 정확한 사이트 명이 기억나지 않을 경우 임의의 명사를 입력할 수 있으며, 이때에는 해당 명사를 전문으로 담당하는 추천 사이트를 추론엔진에서 대체하게 된다. 마지막으로는 구체적인 타겟 사이트 명을 지정하는 경우이다.

"행위" 구성요소에는 수많은 웹사이트에서 제공하는 서비스들에 대해 크게 단순 방문, 키워드 검색, 목적대상 비교, 목적대상 조회 등과 같은 행위들이 존재한다. 물론 서비스의 성격에 따라 행위들은 지속적으로 추가된다.

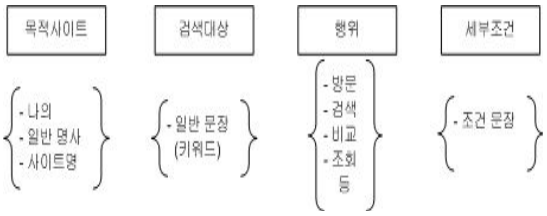

그림6. 구성요소의 예시
Fig6. samples of subquery elements

다음은 목적사이트의 내용에 따른 각각의 예시이다.

경우1: 나의 웹 메일들을 확인하라
(. 나 . 웹 메일 . 확인)
경우2:특허사이트에서 유비쿼터스에 대해 검색하라.
이 경우 특허 전문 사이트로 자동
이동하게 됨.
(. 특허 . 유비쿼터스 . 검색)
경우3: 네이버에서 유비쿼터스란 무엇인가에 대해
조사하라.
(. 네이버 . 유비쿼터스란 무엇인가 . 조사)

3.2 사이트/서비스 메니저

닷큐어리 메니저를 통해 추출된 목적사이트, 검색대상, 행위, 세부조건 항목들에 대해 아래 그림7 과 같이 타겟 URL을 생성하는 작업을 담당하는 사이트/서비스 메니저에 대한 시스템은 다음과 같다

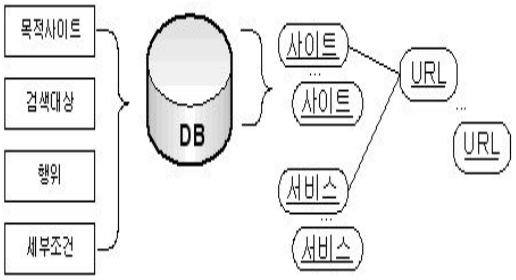

그림7. 타겟 URL을 생성하는 작업
Fig7. Creation Flow of Target URL

입력된 목적사이트, 검색대상, 행위, 세부조건 항목에 대한 최적의 해답을 데이터베이스를 통해 최적의 사이트 및 해당 사이트에서 제공하는 서비스 정보를 추출한다. 추출된 정보에서 입력 받은 항목 중 적정한 내용을 파라미터 값으로 대체하여 최종 타겟 URL 을 생성한다. 이렇게 생성된 URL은 브라우저를 통해 실제 서비스를 받게 된다. 여기에서 복수개의 타겟 URL이 생성될 수 있으며 동시에 복수개의 브라우저를 통해 동시 방문이 가능하도록 구성한다. 사용자가 정의한 내용이 복수개의 결과 값을 가지고 있는 경우 사이트/서비스 매니저를 이용하면 보다 신속하게 웹 페이지에 접근할 수 있다. 만약 사용자가 3개의 웹 메일을 가지고 있다면, 본 논문에서 제시한 검색 에이전트를 이용하여 "나의 웹 메일을 확인하라" 라고 입력하면 다음8 과 같은 결과를 얻을 수 있다.

그림8. 웹 메일 3개를 생성한 결과 창
Fig8. The windows of the created 3 web-mails

3.3 후킹 메니저

사용자가 범용 브라우저에서 임의의 사이트를 방문하고 서비스를 수행할 경우 브라우저 내부에서는 어떠한 프로토콜(HTTP, HTTPs)을 쓰며 어떠한 포트를 사용하는지, 또한 서비스를 수행하는 URL 이 GET 방식인지 POST 방식인지, 파라메타는 어떠한 파라메타를 대체해야 할지를 파악하기 위해 현재 수행중인 브라우저의 내부 데이터(URL)를 후킹 하여야 한다. 이러한 목적을 달성하기 위해서의 소요기술은 Browser Helper Object를 생성하는 기술, IE용 ToolBand 생성 기술, COM 기술을 이용하여 원하는 데이터를 추출한다. 이렇게 하기 위해서는 범용 브라우저인 경우에는 플러그인 모듈을 탑재하여야 한다. 또한 그림9 과 같이 닷큐어리로 시작하는 경우에는 닷큐어리 메니저로 사용자 질의어를 전달하여 형태소 분석을 통한 목적사이트, 검색대상, 행위, 세부조건 항목을 추출하고 사이트/서비스 메니저는 이 값을 기초로 하여 타겟 URL을 생성한 후 다시 후킹 메시지에 해당 URL로 방문을 하도록 지시한다.

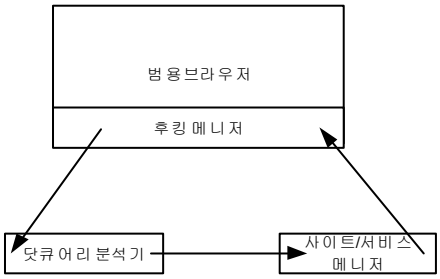

그림9. 후킹 메니저, 닷큐어리 분석기 와 사이트/서비스 메니저 관계
Fig9. Hooking Manager, Dot Query Analyser
and Site/Service Manager

정확한 닷큐어리 분석이 이루어지기 위해서는 사이트/서비스 메니저가 관장하는 데이터베이스의 주기적인 업데이트가 필수적이며, 이를 위한 업그레이드 전용서버 구축 및 버전관리, 그리고 미등록 사이트 및 서비스를 등록하기 위한 사용자 정의 루틴 등이 필수적이다. 본 논문에서는 사용자 정의 루틴은 포함한 반면 자동 데이터베이스 업그레이드에 대한 부분은 고려하지 않았다.

3.4 Job 스케줄러

사용자가 임의로 설정한 시간대에 자동으로 해당 서비스를 수행하고자 할 경우 Job을 생성하고 관리 할 수 있다. 설정하는 방법은 다음과 같이 세 가지 경우가 가능하다.

경우 1: 부팅 후 얼마 있다가 수행
경우 2: 특정 시간에 한번만 수행
경우 3: 매 지정한 시간마다 수행

Job은 여러 개 생성될 수 있고, Job 의 수행과 매치되는 브라우저를 Job 마다 지정할 수 있다. 지정하지 않으면 디폴트로 가용 브라우저가 담당한다. 이 기능은 화면 분할 기능이 가능한 전용 브라우저에서는 강력한 기능이며, 범용 브라우저 사용자에게는 필요시에 복수개의 브라우저를 독립적으로 생성하여 수행할 수 있다.

Ⅳ. 제안 대화형 검색 에이전트의 활용 방법

4.1 제안 에이전트의 화면 구성과 형식

그림10. 대화형 검색 에이전트의 화면 구성
Fig10. Screen of Interactive Search Agent

대화형 검색 에이전트의 구성은 기본적으로 4부분으로 나누어진다. 첫째, 화면 중간에 사용자가 의도하는 문자열을 입력 받고 표시하는 입력 창이 있다. 둘째, 최대 4개의 브라우저를 에이전트로 선택하는 브라우저 에이전트 선택창이 있다. 셋째, 인터넷 서비스를 표시하는 서비스 디렉토리 창이 있다. 마지막으로, 인터넷 서비스 수행 결과를 표시하는 결과 표시 창으로 나누어진다. 서비스 디렉토리 창에는 인터넷 서비스와 해당 서비스를 제공하는 웹사이트가 서로 연결되어 있다. 예를 들어, 정보검색이라는 서비스에는 사용자가 자주 사용하는 "야후"나 "라이코스" 와 같은 웹사이트를 연결시킬 수 있으며, 메일에는 사용자의 메일을 연결시키는 기능을 제공한다. 사용자가 웹 메일을 여러 개 가지고 있는 경우에도 서비스 디렉토리 창에 연결하여 3장의 그림8과 같이 한 페이지에 사용자의 여러 웹 메일을 확인할 수 있는 기능을 제공한다. 또한, 인터넷 서비스를 대, 중, 소 분류하여 서비스 디렉토리 창을 계층적 구조로 구성하는 것도 가능하다. 대분류는 정보 검색, 전자상거래, 메일, 기타 서비스로 분류될 수 있다. 정보 검색은 다시 단어검색, 전자상거래를 위한 상품정보 검색, 가격비교, 언어사전, 뉴스, 여행정보, 각종 코드 검색 등으로 세분할 수 있다. 분류된 언어사전은 또한 영한, 한영, 일한, 중한 등으로 더욱 세분화할 수 있다.

4.2 제안 에이전트를 활용한 예시

우선, 일반 브라우저와 대화형 검색 에이전트의 기능을 비교하면 표1 과 같다.

표1. 일반 브라우저와 대화형 검색 에이전트 기능 비교
Table1. Comparison Table of General Web Browser and Interactive Search Agent

작업 내용	일반 브라우저	대화형 검색 에이전트	비고
OA 작업 중 인터넷 검색	1.현재 작업 중인 프로그램을 최소화한다. 2.브라우저를기동한다. 3.원하는 사이트 주소 입력을 한 후 웹 페이지가 로딩완료 될 때까지 대기한다. 4.검색어 입력 후 원하는 정보 열람한다. 5.정보 열람 완료 후 브라우저를 종료 혹은 최소화한다. 6. 작업 중이던 프로그램을 이전 상태로 복구한다.	1. 상단에 있는 입력 Bar의 주소 입력 창에 원하는 검색어를 입력한다. 2. 열람 후 마우스를 브라우저 외부로 이동하여 자동으로 에이전트를 Hide 시킨다.	6단계에서 2단계로 단순화.
인터넷 전문 검색	1.브라우저를 기동한다. 2.원하는 검색 사이트로 이동한다. 3.검색하고자 하는 단어를 키보드로 입력하거나 클립보드를 활용하는"Ctrl"-C 와 -V를 이용한다. 4.3의 과정을 반복한다.	1.상단에 있는 입력 Bar의 주소 입력 창에 원하는 검색어를 입력하거나 문자열을 영역선택 한 후 마우스 드래그 드롭 한다. (Hotkey를 수행해도 됨)	4단계를 1단계로 단순화.
웹 메일 확인 절차	1.브라우저를기동한다. 2.원하는 웹 메일 주소를 입력하여 이동을 한 후 웹 페이지가 로딩 완료 될 때까지 대기한다. 3.아이디 및 패스워드를 입력한다.	1.어떠한 작업을 수행 중이더라도 상단의 Show 버튼 위에 마우스를 위치하여 에이전트를 자동으로 표시한다. 2.디렉토리에 이미 등록되어 있는 리스트 중 원하는 웹 메일 사이트를 더블 클릭한다.	3단계에서 2단계로 단순화.
복수 브라우저 실행	1.원하는 개 수만큼 브라우저를 기동한다. 2.브라우저 위치와 크기를 각각 조정한다. 3.각 브라우저 별로 원하는 사이트로 방문한다. 4.화면에 실행중인 브라우저별로 일일이 이동을 반복한다. 5.정보 취득 후 현재 실행 중인 브라우저를 개별적으로 종료 혹은 최소화 시킨다.	1.어떠한 작업을 수행 중이더라도 상단의 Show 버튼 위에 마우스를 위치하여 에이전트를 자동으로 표시한다. 2.2"4분할 창을 이용하여 마우스 위치 조정만으로 복수개의 사이트를 방문한다. 3.열람 후 마우스를 브라우저 외부로 이동하여 자동으로 에이전트를 Hide 시킨다.	5단계에서 3단계로 단순화.

사용자가 3장의 예1에 대한 정보 결과를 얻기 위해 일반 브라우저를 이용한 검색 경로를 살펴보면 다음과 같다. 먼저, www.naver.com을 방문하고 검색 입력박스에 유비쿼터스를 Fill-In 하고 "검색" 버튼을 누른다[7,8]. 이것은 search.naver.com 사이트의 CGI 루틴에 적정한 파라미터를 세팅하여 수행되며, 실제적인 수행 URL은 아래와 같다.

http://search.naver.com/search.naver?where=nexearch&query=%C0%AF%BA%F1%C4%F5%C5%CD%BD%BA&frm=t1&sm=top_hty

물론 서비스마다 단순 Html을 표시할 수도 있고, GET 방식의 URL 과 POST 방식일 수도 있으며, Secure HTTP 일 경우도 있을 것이다[9,10,11].

위의 예를 본 논문에서 제시한 검색 에이전트의 입력 창을 이용해 수행하면 다음과 같다.

그림11. 검색 에이전트의 수행
Fig11. Search Agent Operation

그리고 검색 에이전트에서 실행한 후에 결과 표시 창에 나타난 검색 결과는 다음과 같다.

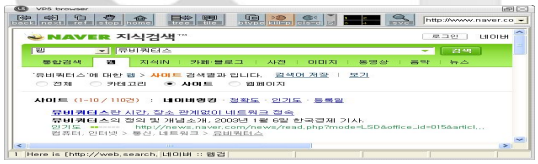

그림12. 검색 에이전트의 결과 창
Fig12. Searched Result Screen of Search Agent

4.3 제안 에이전트의 재방문 시 접근 방법

하나의 웹 페이지에서 시작되는 관련된 웹 페이지들의 링크 수는 페이지 당 수십 개 이상으로 이루어져 정보 검색과 재방문의 접근에 영향을 준다[13,14,15]. 본 논문에서는 제시한 검색 에이전트의 서비스 디렉토리 창을 이용하여, 사용자가 지정한 웹 페이지를 웹사이트 내의 메인 페이지로 정하여 재방문을 했을 경우 불필요한 이동 횟수를 줄일 수 있다. 그리고 대부분의 인터넷 사용자들은 1개 이상의 웹 메일들을 가지고 있어, 이런 메일들을 일일이 방

문하여 메일을 확인하는 절차를 거치고 또 다른 메일도 이와 같이 확인하는 절차를 거쳐야 한다[16]. 그리고 재방문 시에도 같은 절차를 반복해야 한다. 하지만 본 논문에서 제시한 검색 에이전트에서는 자주 사용하는 웹 메일의 URL들을 한 페이지에 등록하여 한 번에 여러 개의 메일을 확인하는 기능을 제공하고 있다[그림8].

다음은 사용자가 즐겨 찾는 사이트가 Google, Paran, Yahoo 그리고 Nate일 경우에 본 논문에서 제안한 검색 에이전트를 이용하여 구현된 결과를 나타낸다.

그림13. 검색 에이전트로 구현된 자주 사용하는 사이트
Fig13. Frequent Using Site by Search Agent

위에서 알 수 있듯이, 즐겨 찾는 사이트들을 일일이 방문하여 검색하는 절차를 본 논문에서 제시한 검색 에이전트는 한 페이지에 저장하여 정보를 검색하는 기능을 제공한다.

다음은 "웹"에 대한 검색을 본 논문의 검색 에이전트를 이용하여 구현된 결과를 나타낸다.

그림14. 검색 에이전트로 구현된 "웹"에 대한 검색
Fig14. The Search of "Web" word by Search Agent

다음 그림15는 전용프로그램과는 별도로 범용브라우저(IE)내에서의 닷큐어리를 처리할 수 있는 구현물로, IE내에 플러그 인되어 있는 DotQuery 툴바에 대한 내용으로 ".네이버.유비쿼터스.검색"에 대한 결과 내용을 담고 있다.

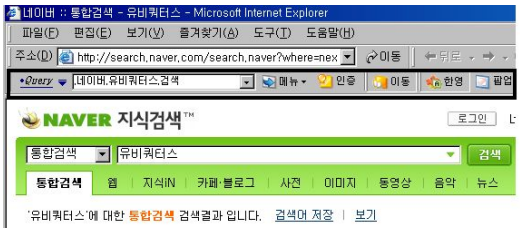

그림15. IE용 닷큐어리 툴바
Fig15. DotQuery Toolbar of Internet Explorer(IE)

4.4 속도개선 결과 및 효율성

닷큐어리를 활용한 대화형 검색 에이전트의 활용으로 인해 많은 속도 절감 및 무의미한 네트워크 트래픽 감소로 인해 효율적인 인터넷 사용을 구현하였다.

아래의 표2는 일반적인 방법으로의 웹사이트 방문의 경우와 닷큐어리를 활용할 경우의 속도 개선에 대한 비교표이고 표3은 두 가지 방법에서의 네트워크 트래픽의 양에 대한 결과표이다. 속도 및 트래픽 측정의 근거는 다음의 경우이다.

<일반적인 방법>
스텝1: 사용자가 브라우저에서 "네이버"를 방문한다.
스텝2: 키워드 박스 내에 "유비쿼터스"를 입력한다.
스텝3: "검색" 버튼을 눌러 키워드 검색을 수행한다.

<닷큐어리를 활용>
스텝1: 주소창(그림11) 혹은 툴바주소창(그림15)에서 ".네이버.유비쿼터스.검색"을 입력한다.

표2. 속도비교 테이블
Table2. Speed Comparison

방 법	작업 내용	소요시간
일반적인방법	1.네이버 방문	3.5 초
	2.키워드박스내에 "유비쿼터스" 입력	3.0 초
	3.검색수행 및 결과	2.0 초
	합 계	8.5 초
닷큐어리 활용	1. 주소창에 ".네이버.유비쿼터스.검색" 입력	4.5 초
	2. 검색수행 및 결과	2.0 초
	합 계	6.5 초

표3. 네트워크 트래픽 테이블
Table3. Traffic Comparison

방 법	작업 내용	트래픽
일반적인방법	1.네이버 방문	248 Kb
	2.키워드박스내에 "유비쿼터스" 입력	0 Kb
	3.검색수행 및 결과	495 Kb
	합 계	743 Kb
닷큐어리 활용	1. 주소창에 ".네이버.유비쿼터스.검색" 입력	0 Kb
	2. 검색수행 및 결과	495 Kb
	합 계	495 Kb

위의 표2와 같이 일반적인 방법에 비해 닷큐어리를 활용할 경우 인터넷 활용 속도는 24% 개선할 수 있었다. 물론 사용자 컴퓨터 사양과 인터넷회선상태, 타이핑 속도에 따라 약간의 차이가 발생할 수는 있지만 속도차이의 가장 큰 원인은 검색을 하기 위한 선행 작업으로 사이트 방문이 필수적이고 그로 인해 불필요한 초기화면이 표시될 때까지 기다려야 한다는 것이다. 위의 경우보다는 복수의 웹사이트 방문을 병렬로 수행할 경우에 닷큐어리의 효용성은 더욱더 커지게 된다. 즉 n 개의 사이트에서 검색을 순차적으로 수행할 경우보다 닷큐어리를 활용하여 병렬검색을 수행할 경우 n X 0.24% 만큼 속도 향상을 기할 수 있게 된다.

여기서 또 다른 중요한 효율성은 무의미한 사이트 초기 화면 스킵으로 인한 네트워크 트래픽 감소를 들 수 있다.

위의 표 3에서와 같이 닷큐어리를 활용할 경우 일반적인 방법보다 네트워크 트래픽의 양을 약 34% 절감할 수 있게 되어 그 만큼의 네트워크 오버헤드를 줄일 수 있었다.

V. 결론 및 향후연구

본 논문에서는 인터넷 사용자들이 웹에서 원하는 정보와 서비스를 획득하기 위해 반드시 선행되어져야 할 여러 단계의 프로그램 조작 절차를 단순화시키고, 수많은 사이트마다 고유의 서비스 구조들을 모르는 상태에서도 원하는 서비스를 쉽게 취득할 수 있는 방법에 관한 문제를 다루었다. 또한 이러한 문제 해결을 위해 닷(.)으로 시작하는 자연어 기반 질의어(닷큐어리)를 활용하는 대화형 검색 에이전트를 제안하였다. 닷큐어리의 유형중 자연어 기반의 닷큐어리에, 보다 더 정확한 정보를 접근하기 위해 서브 닷큐어리를 이용하였고, 이러한 서브 닷큐어리를 활용함으로써 불필요한 컴퓨터 자원 소모를 줄일 수 있고 사용자의 정확한 의도를 파악할 수게 한다.

본 논문에서 제안한 방법으로 생성된 대화형 검색 에이전트를 사용함으로써, 웹사이트에서 제공되는 다양한 서비스 접근에 소요되는 시간 절감 및 네트워크 트래픽 감소에 따른 인터넷 사용 효율성이 증대된다.

컴퓨터 미숙련자들을 위한 인식률이 매우 높은 자연어기반의 닷큐어리 지원에 대한 시스템 구현이 향후 연구과제로 남아 있다.

참고문헌

[1] "Web Accessibility Initiative(WAI)", http://www.w3.org/WAI/.

[2] Ning Zhong, Jiming Liu, Yao, Y.Y., Ohsuga S., "Web Inteligence(WI)", Computer Software and Applications Conference, 2000, COMPSAC 2000, The 24th Annual International 2000, pp. 469-470, 2000.

[3] Divyakant A., Wen-Syan L., Quoc V., "Retrieving and Organazing Web Pages by Information Unit", 10th www, May 1-5 2001, Hong Kong., ACM 1-58113-348-0/01/0005.

[4] Minqing H.,Xiaoli Li, Bing L., Tong-Hong P., "Web Search Based on Micro Information Units", 11th International WWW Conference, 2002.

[5] Shaun K., "Integrating Back, History and Bookmarks in web Browsers", Proceedings of CHI'01, ACM Press, pp.379-380, 2001.

[6] Teruhisa M., Tsuyoshi E, Seiji I., "Fast Web by Using Updated Content Extraction and a Bookmark Facility", The 4th International ACM conference on Assitive technologies 2000.

[7] J. wing, and Kleppe, A., "OCL : The Constraint Language of the UML", JOOP, May, 1999.

[8] T. Milo and D. Suciu, "Index Structures for Path Expressions", Proc. Int'l Conf. On Database Theory(ICDT), pp.277-295, 1999.

[9] 이경순, 김재호, 최기선, "한국어 질의응답시스템에서 자료 유형에 따른대답검색 및 대답해석", 2001년 한국인지과학회 춘계학술대회, pp.73-78, 2001.

[10] 이승익, 조성배, "웹기반 대화형 에이전트", 정보과학회논문지, 9권, 5호, pp.530-540, 2003.

[11] Bill Venners, "Inside the JAVA 2 Virtual Machine", McGrawHill, 2000.
[12] "Web Services Activity ", http://www.w3.org/Consortium/activities#URIActivity
[13] 정재목, 김형주, "웹 정보의 추출 및 통합을 위한 래퍼 시스템", 정보과학회논문지, 제9권, 제5호, pp.551-559, 2003.
[14] G, Huck, P. Fankhauser, K. Aberer, and J. Neuhold. Jedi, "Extracting and synthesizing information from the web", In CoopIS 1998, pp32-43, 1998.
[15] 이덕근, 오미화, 조재훈, 최인수, "DW에서의 질의어 처리 성능향상을 위한 데이터 구조화 방법", 한국컴퓨터정보학회 논문지, 제10권, 제1호, pp.7-13, 2005.
[16]] T.BernersLee, R.Fielding and., Masinter, "Uniform Resorce Identifiers(URI):Generic Syntax", http://www.ietf. org/rfc/rfc3986.txt., 2005.

저 자 소 개

김선옥

1991 서강대학교 대학원 이학석사
1998 서강대학교 대학원 이학박사
1999-2004 (주)맥스미디어코리아 개발이사
2005-현재 한라대학교 정보통신공학부 전임강사
<관심분야> 멀티미디어 시스템, 정보 통합, 정보검색