

# 통합검색시스템: Federated Searching



심 경

정보학박사  
한국도서관협회 평생회원  
(주)아이리스넷 대표  
shim@irisnet.co.kr

도서관에서 통합검색이라고 하면 가장 먼저 생각나는 것은 Z39.50이다. 한 지인이 Z39.50라는 말을 처음 듣고 이게 “제트39.50”이나 “지39.50”이 맞느냐를 가지고 술집에 앉아 친구들과 논란을 벌였다고 한다. 이 논란을 평정한 한 친구의 발언은 “야, 마징가Z가 마징가 제트지, 마징가 지나?”라는 일갈이었다. 발음이야 어떻든 우리가 배운 것 중 기억에 남는 것은 “다수의 도서관시스템을 한꺼번에 검색할 수 있는 프로토콜”이라는 정도이다. 문득 요즘 도서관 서비스는 “소장”을 넘어 “접근”이라는 패러다임에 따라 다수 도서관을 통합검색하고 싶어하고 단일 도서관 내에서도 많은 시스템과 각종 데이터베이스가 산재해서 검색이 어렵다는데, “그걸 쓰면 되지 않나?” 또는 “왜 그걸 쓰면 안될까?” 하는 사람이 있을 것이다.

그 해답으로 Z39.50은 부분해결책을 제공하지만 완전한 답을 제공하지 않는다는 것이다. 대신 Federated Searching이라고 부르는 통합검색시스템이 몇 년 전부터 등장해 상당히 인기를 끌었다. 국내에도 한 업체가 보급하여 거의 모든 대학도서관에 독점 납품을 하다시피 했던 메타서치(MetaSearch)라는 제품이 있었다. 당시 복잡한 도서관서비스 중 원하는 자료를 찾기 위하여 어떤 시스템 또는 데이터베이스를 검색해야 하는가라는 이용자의 고민을 해결하고 싶어하던 도서관들 사이에 인기몰이를 하여 옆에서 보기에는 “대박”을 터뜨린 것처럼 보일 정도였다. 이 시스템은 거의 모든 대상을 통합하여 검색하고 결과를 제시한다고 하니 너무나 신기하여 마술같이 보일 정도였는

데, 문득 영어의 “현실이라기엔 너무 좋다!(Too good to be true!)”라는 표현이 떠오르며 뭔가 있지 않을까 하는 의심이 생겼던 것도 사실이다.

일전에 필자가 “벤더들은 거짓말을 하지 않는다. 다만 모든 것을 말하지 않을 뿐이다”라는 말을 한 적이 있다. 벤더는 원리를 설명하는 것이 아니라 주로 제품의 기능과 인터페이스만을 보여주기 때문이며, 벤더 측에서 설명하지 않는 것을 우리가 미리 알기 위해서는 항상 해당 솔루션의 기본 원리를 이해하고 이를 기반으로 발생 가능한 문제점을 도출하여 직접 질문을 하는 것이 가장 바람직하다. 그러므로 이 글에서는 통합검색시스템이 무엇이며, 그 기본 원리는 어떻게 되고, 또 이와 같은 솔루션에 감추어진 문제점은 없는가를 살펴보자.

### 통합검색시스템이란 무엇인가?

사실 “통합검색시스템”이란 용어는 널리 쓰이는 일반명사이므로 이 글에서와 같이 특정 기법을 지칭하는 이름으로 적절하지는 않다. 이 명칭은 우리 분야에서 협의되고 결정되어 받아들여지는 명칭이 아니라 필자가 우리말화한 것일 뿐으로, 지칭되는 대상에 대하여 오해의 소지가 있다. 왜냐하면 통합검색기법은 크게 나누어 다수 데이터베이스의 통합색인을 구축하는 방식과 두 시스템 간에 API라는 공통 인터페이스를 제공하는 방식<sup>1)</sup>, 그리고 대상 데이터베이스의 웹 검색 인터페이스를 분석하여 수행하는 방식의 세 가지가 있기 때문이다. 이 글에서 통합검색시스템이란 후자의 두 가지 방식을 병합한 것을 의미하며, 이 제품들은 Z39.50, XML Web Services, HTTP 등 사용 가능한 모든 프로토콜과 스크린 스크래핑(screen scraping) 또는 페이지 스크래핑(page scraping) 기법을 활용하여 통합검색을 가능케 한다. 이 기법을 영어로는 흔히 federated search, meta search, single search 또는 cross search라고 한다. 종종 이들 뒤에 시스템 또는 소프트웨어라는 단어가 함께 와서 해당 솔루션에 대한 일반호칭으로 쓰이기도 하고 search뒤에 ing를 붙여서 그 기법 자체를 지칭하기도 한다. 또 어떤 이들은 multiple database searching, polysearching, broadcast searching이라고 표현하기도 한다.

통합검색시스템의 기본기능은 다수의 데이터베이스 또는 시스템을 단일 인터페이스에서 하나의 질의문으로 동시에 검색하고 검색결과 또한 단일화면에 나열해 주는 프로그램이다. 그 장점은 동일검색을 여러 시스템을 대상으로 순차적으로 반복하는 불편함을 없애주는 것은 물론이

1) Application Programming Interface(API)는 외부에서 자체 데이터베이스에 접근할 수 있도록 “통로”가 되는 프로그램을 의미한다. 도서관 환경에서는 타 시스템의 접근 및 검색기능을 제공하는 측에서 구현하거나 W3C Web Services나 Z39.50와 같은 표준적 스펙을 구현한 것이 있다. 엄밀히 말하여 Z39.50는 검색프로토콜이며 API는 아니지만, 이 글의 전개 목적 상 세 가지 방식 중 나머지 두 가지는 아니며 연계대상 두 시스템 간 그 무엇을 구현해야 한다는 점을 들어 API 범주로 취급하였다.

고 복수시스템의 검색결과가 단일 인터페이스에 제공되므로 원하는 자료를 선택하여 수집하는데도 편리하다는 것이다. 단일 인터페이스를 통한 검색 및 결과제시라는 기능 이외에 이들 제품은 주로 여러 데이터베이스에서 검색된 결과를 통합하고(merge), 중복제거(de-duplicate)한 후, 결과를 정렬(sort)하거나 적합성에 따라 순위화(relevancy rankings)하여 제공한다. 사실 도서관환경에서 복수도서관시스템을 동시에 검색하기 위하여는 1970년대 초기부터 준비하여 1988년에 초기 버전이 발표된 Z39.50이 있는데<sup>2)</sup> 왜 그것은 부분해결책만을 제공하고, 통합검색시스템이 그 이외의 다른 기법을 “통합”해서 사용하는 것일까? 우선 Z39.50의 특성과 구현상황을 살펴보자.

Z-프로토콜은 데이터베이스 검색을 허용하는 측에 Z-서버를, 그리고 이를 검색하려는 측은 Z-클라이언트를 Z39.50 명세서에 따라 구현해야 하는데, 이를 구현하지 않은 기관이 구현한 기관보다 훨씬 많다. 또한 Z-프로토콜은 이론적으로는 잘 작동하도록 되어있고 기능이 복잡하나 실제 사용기관에 일관성 있게 적용되지 않고, 또 구현하였더라도 외부접근을 허용하지 않는 경우가 있다<sup>3)</sup>. Z-프로토콜은 웹 이전의 기술로서 W3C의 Web Services라는 웹 상에서 프로그램 간의 연계를 위한 표준 인터페이스가 제시되었지만 아직은 국내외를 막론하고 이를 지원하는 도서관시스템은 극소수이다<sup>4)</sup>. 그 밖에 다른 프로토콜을 사용하는 기관도 있고 그 프로토콜이 무엇인지 알 수 없는 기관도 있다. 이러한 상태에서 다수의 시스템을 단일 인터페이스로 검색하고자 상업분야에서 발전한 방식이 스크린 스크래핑 또는 페이지 스크래핑이라는 기법이다. 이 기법 자체에 대하여는 뒤에 좀더 상세히 설명한다.

### 통합검색은 어떻게 작동하나?

통합검색시스템은 자칫 통합색인구축에 의한 통합검색(이하 “통합색인방식”)과 혼동되는 경우가 있는데, 통합색인방식은 그림 1과 같이 대상 자원에 대한 단일 마스터 색인(master index)을 구축하여 복수시스템을 동시에 검색이 가능하도록 하는 것이다. 이 방식의 장점은 단일색인이 구축

2) Lynch, C. A. (1997). The Z39.50-1995 Information Retrieval Standard: Part I. A Strategic View of Its Past, Present and Future. D-Lib Magazine, April. Retrieved June 24, 1998, from <http://www.dlib.org/dlib/april97/04lynch.html>

3) Z39.50는 클라이언트와 서버로 구성되며 상대 시스템을 검색하려면 검색하는 측은 Z-클라이언트가 있어야 하고, 검색을 지원하는 측은 Z-서버를 갖추고 있어야 한다. Z39.50 프로토콜의 복잡한 기능 내에서 각 기관의 구현내역은 다양하다. 이와 같은 구현내역을 프로파일이라 하며, Z-서버 측은 이 중 연결에 필요한 변수를 설명하는 페이지를 일반적으로 자관 홈페이지 한 곳에 공개하고 있다. 이것이 “일관성 있게 적용되지 않고”라는 부분의 의미이며, 결과적으로 표준검색프로토콜이지만 연계를 위하여 Z-클라이언트는 기관마다 별도로 세팅을 해주어야 한다는 것을 의미한다. 또한 “외부접근을 허용하지 않는 경우”란 Z-클라이언트만을 구현하여 Z-서버를 제공하는 타 기관 검색은 하지만, 자체 Z-서버를 서비스 하지 않는 경우가 있고, Z-프로토콜을 목록시스템에서 서지유틸리티를 대상으로 서지레코드 검색 및 반입/반출을 위하여 사용하는 경우도 이에 포함된다.

4) 국내 전자도서관시스템 중 Web Services를 지원하는 것은 한국과학기술정보연구원(KIST)의 KIRIS가 유일하다.

되므로 검색속도가 빨라 이용자가 기다리는 시간이 짧아진다. 하지만 대상 시스템의 모든 내용에 대한 색인을 구축하기 위하여 검색엔진이 모든 자원의 문서나 데이터베이스에 접근해야 하므로 “슈퍼 사용자(super user)” 권한을 부여해야 한다는 문제가 발생한다. 슈퍼 사용자 권한 부여는 재무나 인사정보와 같은 민감한 정보에도 접근하게 되는 보안문제를 야기할 수가 있으므로 이 방식은 주로 동일 기관의 다수 시스템이나 정책적으로 관련된 기관 간의 통합검색에만 주로 사용될 수 있다는 제한이 있다.

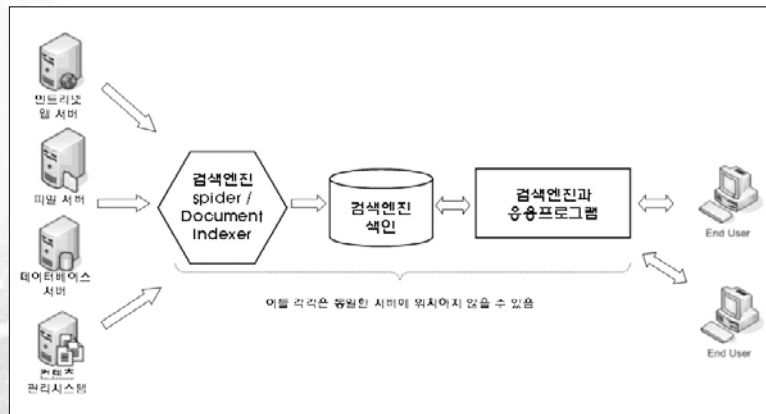


그림 1. 통합색인 구축에 의한 통합검색

하지만 복수도서관이나 타 기관의 자료검색을 위하여 통합색인을 구축하는 것은 기술적으로는 가능할 지라도 정책적으로 가능하지 않다. 그러므로 상대가 허용하는 범위 내에서 통합검색을 수행하는 방법이 필요하며 이를 만족시키는 기법이 앞서 통합검색시스템이라고 칭한 federated searching이다. 통합검색시스템은 앞서 설명한 바와 같이 상대기관의 시스템에서 사용하는 가능한 모든 프로토콜을 활용하여 다수 대상에 동시에 검색문을 보내 검색하고 결과를 하나의 인터페이스로 통합하여 제공하는 방식을 사용한다. 다시 말하여 만약 대상 기관이 Z 프로토콜이나 W3C의 XML Web Services 와 같은 표준검색프로토콜이나 API를 채택하고 있으면 이를 활용하고, 이와 같은 표준의 사용이 가능하지 않은 대상은 소위 말하는 스크린/페이지 스크래핑 기법을 사용한다. 스크린/페이지 스크래핑이란 표현 그대로 접속할 대상의 URL과 HTML스크린/페이지를 분석하여 해당 페이지에서 사용되는 검색 및 검색결과 구문을 분석하여 이들에 대한 프로파일(profiles)을 구성한 후 검색 시 활용하는 것이다. 이 분석방법에서 검색의 구문(query's syntax)을 분석하는 방법은 상대기관 검색페이지의 “소스” 또는 “속성”을 선택한 후 해당 소스를 보며 시행착오를 거쳐 발견하며, 검색결과 페이지도 동일한 절차를 거친다(그림 2 참조). 이는 기술적으로 대단히 어렵지는 않지만<sup>5)</sup> 상당한 시간이 소요되어 프로그래머들 사이

에서는 “노가다”라는 표현이 사용되기도 한다. 그러므로 검색대상기관은 자신의 시스템에 대한 설명을 하거나 노력을 기울일 필요가 전혀 없다. 국내의 경우 대개 상대기관은 자신이 검색대상이 되는지도 모르는 경우가 대부분이다<sup>5)</sup>.

선택 번호	서명	제자	압축서	압축년
<input type="checkbox"/> 1	사람의 권리, 中	오이찬순장 (표창) [해]	SearchItem = 1818M, &nb	1974
<input type="checkbox"/> 2	이2년법 : 인간의 대지 / 인간 조건	Saint-Esprit Antoine de	SearchItem = 1818M, &nb SearchKey = 1818M SearchMethod = 1818M SearchInformation = 1818M SearchAuthority = 1818M	
<input type="checkbox"/> 3	사람의 대지 : 어진 至手 : 東方郵便報 : [海] / 人間의 대지	Saint-Esprit Antoine de	SearchItem = 1818M, &nb SearchKey = 1818M SearchMethod = 1818M SearchInformation = 1818M SearchAuthority = 1818M	
<input type="checkbox"/> 4	사람의 권리, 3	오이찬순장 (표창) [해]	SearchItem = 1818M, &nb SearchKey = 1818M SearchMethod = 1818M SearchInformation = 1818M SearchAuthority = 1818M	
<input type="checkbox"/> 5	사람의 권리, 下	오이찬순장 (표창) [해]	SearchItem = 1818M, &nb SearchKey = 1818M SearchMethod = 1818M SearchInformation = 1818M SearchAuthority = 1818M	
<input type="checkbox"/> 6	이2년법 : 인간의 대지 / 인간 조건	Saint-Esprit Antoine de	SearchItem = 1818M, &nb SearchKey = 1818M SearchMethod = 1818M SearchInformation = 1818M SearchAuthority = 1818M	
<input type="checkbox"/> 7	스칸의 사할 / 人間의 天地 / 人間의 대지	Proust, Marcel	SearchItem = 1818M, &nb SearchKey = 1818M SearchMethod = 1818M SearchInformation = 1818M SearchAuthority = 1818M	
<input type="checkbox"/> 8	사람의 권리, 海部-第2部	오이찬순장 (표창) [해]	SearchItem = 1818M, &nb SearchKey = 1818M SearchMethod = 1818M SearchInformation = 1818M SearchAuthority = 1818M	
<input type="checkbox"/> 9	인간의 조건	Arendt, Hannah	SearchItem = 1818M, &nb SearchKey = 1818M SearchMethod = 1818M SearchInformation = 1818M SearchAuthority = 1818M	
<input type="checkbox"/> 10	사람의 권리, 2	오이찬순장 (표창) [해]	SearchItem = 1818M, &nb SearchKey = 1818M SearchMethod = 1818M SearchInformation = 1818M SearchAuthority = 1818M	
<input type="checkbox"/> 11	사람의 권리, 中	오이찬순장 (표창) [해]	SearchItem = 1818M, &nb SearchKey = 1818M SearchMethod = 1818M SearchInformation = 1818M SearchAuthority = 1818M	
<input type="checkbox"/> 12	사람의 권리, 2	오이찬순장 (표창) [해]	SearchItem = 1818M, &nb SearchKey = 1818M SearchMethod = 1818M SearchInformation = 1818M SearchAuthority = 1818M	

그림 2. 검색결과화면의 “소스”보기를 활용한 분석 예

이와 같이 여러 종류의 프로토콜을 단일 시스템에서 활용하기 위하여 대상 시스템에 대한 분석내용인 프로파일은 그림 2의 “통합검색기”에 숨겨져 있으며 우리에게 보여지는 것은 검색인터페이스와 결과화면 뿐이다. 따라서 통합검색기가 제공하는 검색인터페이스에서 이용자가 검색문을 입력하면 통합검색시스템은 하나의 검색문을 개별 프로파일에 전달하여 이들이 해당 시스템을 검색하여 결과를 가져오도록 한다. 이와 같이 사이트마다 분석한 프로파일(profile)을 가지고 있는 곳이 그림 3의 “Federated Data Silo”이다. 따라서 이 프로파일은 그림 3의 검색 시스템 1~3과 같이 그 검색프로토콜이 Z39.50, XML Web Services, 또는 HTTP 프로토콜(일반 웹 페이지) 중 어느 것일 수 있으며, 그들 다양한 대상시스템에 대한 분석결과를 저장하고 있는 곳이다. 따라서 통합검색시스템은 검색대상시스템의 수와 동일한 프로파일이 구축되어 저장된다.

통합검색시스템이 앞서 설명한 통합색인방식과 가장 현저한 차이점은 검색을 위하여 미리 통합색인을 구축하지 않고 그림 2에 제시된 바와 같이 개별 벤더시스템에 구축된 다수 색인을 그들이 사용하는 검색 구문을 준수하여 동시에 검색을 하도록 프로파일을 활용하는 것으로 어찌 보면 “깔끔한” 기법은 아니라고 할 수 있다.

5) 물론 경험이 있는 프로그래머에 한해서 해당되는 말이다. 초급기술자가 쉽게 할 수 있는 정도는 아니다.  
6) 그렇다고 보안문제가 발생하는 것은 아니다. 이는 이용자용 웹 검색화면에서 허용하는 범위 내에서만 분석이 가능하며 검색되는 데이터의 범위도 마찬가지이다.

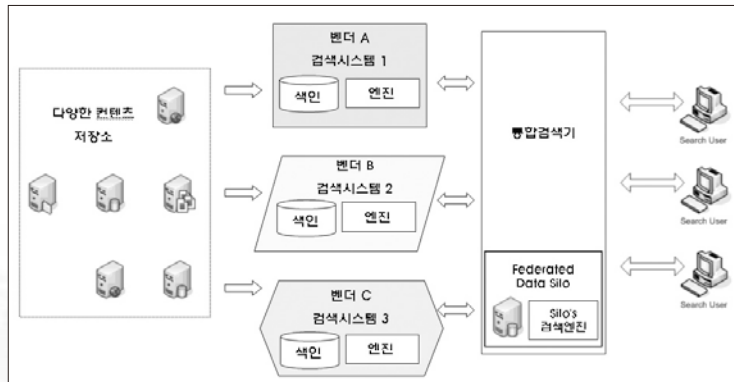


그림 3. 통합검색시스템의 원리

### 통합검색은 만병통치약인가?

앞에서 통합검색시스템은 어떤 종류의 프로토콜을 사용하는 대상이건 모든 시스템을 검색할 수 있고, 결과를 통합하고, 중복을 제거한 후, 결과를 정렬하거나 순위화하여 제공하나, “깔끔한” 기법은 아니라고 하였다. 왜냐하면 사실 앞서 나열한 장점은 일반적으로 벤더들이 주장하는 내용이며, 깔끔하지 못하다는 것은 다음과 같은 질문들에 대한 제한점이 있기 때문이다.

**통합검색시스템은 모든 시스템을 검색할 수 있는가?** 모든 통합검색시스템 제품이 Z39.50을 사용하는 시스템이나 무료 데이터베이스에 대한 통합검색은 가능하겠지만 “모든” 데이터베이스나 시스템을 검색할 수 있는 것은 아니다. 그 이유는 사용자인증(authentication) 절차 때문이며, 일부 벤더들은 이를 솔직히 시인한다. 이에 해당되는 대상으로 상용 전자저널 패키지를 들 수 있다. 이 말에 일부 이용자들은 “우리 학교에서 사용하는 통합검색시스템은 우리가 원하는 타 대학이나 우리가 구독하는 모든 상용데이터베이스를 동시에 검색하는데...”라고 생각할 수 있다. 이 경우는 두 가지 해석이 가능하다. 첫째, 현재 사용중인 통합검색솔루션이 인증절차를 해결하는 제품일 수 있고, 둘째, 교내에서 수행한 검색만을 보았을 때일 것이다. 후자의 경우에 사용자 인증문제가 해당되지 않는 이유가 대개 상용데이터베이스도 계약 시 주로 도서관이나 교내에서 사용하는 이용자를 대상으로 라이선스 계약을 하므로 해당 학교의 IP 대역에서의 접근을 허용한다. 하지만 교외 사용자(remote users)는 전혀 다른 문제를 야기한다. 상용데이터베이스를 교외에서 접속하는 이용자를 지원하는 방법은 단순하지 않으므로 통합검색솔루션을 구매하기 전에 이 부분은 반드시 확인이 필요하다.

**중복제거(de-dupe)는 정말 가능한가?** 비록 서로 다른 시스템 또는 데이터베이스를 검색하더라도 동일한 검색문을 사용하면 검색결과에 중복레코드가 발생하기 마련이다. 쉬운 예로 “정보검색”이라는 검색문을 몇

개의 대학도서관 OPAC에서 검색했다고 가정하면 쉽게 짐작할 수 있다. 더구나 다수 데이터베이스를 동시에 검색하면 검색결과는 방대해질 것이 당연하고 이용자는 그들 중 앞에서 본 레코드를 다시 보고 싶어하지 않는다. 그런데 벤더들은 이러한 중복을 제거한 검색결과를 제시한다고 주장한다. 사실 이는 현실적으로 불가능하며 잘못된 주장이다. 우선 중복제거를 위하여 통합검색시스템은 모든 대상 데이터베이스로부터 검색결과를 다운로드 받고 그들을 비교해야 한다. 하지만 첫째, 대상 시스템이 반응하는 시간은 상대 서버의 사양과 네트워크에 따라 다양하며 이를 검색하는 측에서 바꿀 방법은 없다. 둘째, 데이터베이스는 일반적으로 검색결과를 한번에 10~20레코드씩 제시한다. 그런데 검색결과에 100,000레코드가 포함되었다면 그리고 만약 10~20레코드를 다운로드 하는데 5초가 소요된다면, 이 상황에서 완벽한 중복제거를 하려면 일단 전체 레코드가 모두 다운로드 될 때까지 최소 한 시간 이상을 기다려야 할 것이다. 따라서 벤더가 중복제거를 한다는 말은 단지 검색결과의 초기 세트에 대한 중복제거를 한다는 의미이다.

**적합성 순위화(relevancy rankings)는 정말 적합한가?** 일반적으로 불리안 검색기법에 기반한 시스템은 순위화가 불가능하다. 이들은 집합이론에 근거하며, 집합이론에서는 부분 멤버십(partial membership)을 허용하지 않기 때문이다. 따라서 순위화를 한다고 하면 대상문헌에 검색문에 포함된 용어의 출현빈도를 계산하여 가장 빈도가 높은 문헌을 검색결과리스트의 맨 위로 보내고 점차적으로 낮은 빈도로 나열한다. 기본적인 문제는 이 순위화 기법의 실효성에 있으나 현실적으로 검색대상 데이터베이스가 제공하는 검색대상 필드의 일관성 결여가 사태를 더 심각하게 한다. 즉, 우리가 통합검색시스템으로 초록과 원문(full-text)을 제공하는 데이터베이스를 검색한다고 하더라도 실제 검색대상이 되는 필드는 소위 말하는 서지사항 뿐이다. 왜냐하면 모든 대상 데이터베이스가 초록필드와 원문을 가지고 있는 것이 아니므로, 통합검색시스템은 가장 일반적인 서지사항이나 기사색인데이터를 대상으로 검색하기 때문이다. 또한 순위화도 모든 검색결과가 다 도착해야 가능한 일이다. 따라서 불가능하지는 않을지라도 그 효율과 효과가 만족스럽지 못할 것이다.

**통합검색솔루션은 서비스인가 소프트웨어인가?** 통합검색시스템은 틀림없이 소프트웨어이지만 서비스의 성격을 가진다고 이해하는 편이 옳다. 앞서 이 기법의 기본원리를 설명하는 과정에서 통합검색시스템은 상대기관에게 협조를 구하거나 상대기관이 통합검색시스템을 사용하는 기관에 대한 서비스 의무가 없는 상태에서 대상을 분석하여 통합검색이 가능하도록 하는 기법이라고 하였다. 결과적으로 분석대상 사이트가 서비스를 변경하거나 웹 페이지를 재구성하는 경우에는 통합검색결과에 직접적으로 영향을 미친다. 이는 변경된 기관의 검색결과에서 에러가 발생하며, 흔치는 않으나 이 에러가 전체검색에 문제를 일으키는 경우도 있다. 한 문헌에 의하면 통합검색시스템의 검색대상이 되는 데이터베이스는 연간 2~3차례에 걸쳐 업데이트와 수정이 일어난다고 한다. 이는 만약 통합검색의 대상 데이터베이스가 100개라면 통합검색시스템의 프로파일을 연간 200~300번 정도 수정을 해야 한다는 것을 의미한다. 그러므로 서비스의 중단이나 문제를 즉각 조치 또는

예방하기 위하여 통합검색솔루션은 구매보다는 구독이라는 개념으로 생각하는 것이 중요하다<sup>7)</sup>.

**통합검색솔루션은 현재 사용중인 검색엔진의 기능을 향상시키는가?** 그렇지 않다. 통합검색솔루션은 단순히 대상 데이터베이스에서 제공하는 동일한 콘텐츠를 검색할 뿐 기존 데이터베이스의 검색인터페이스를 향상시키지는 않는다. 통합검색솔루션은 검색문이 입력되면 이를 검색대상 기존 데이터베이스의 엔진이 이해하도록 그것을 통역하는 역할을 할 뿐이며, 그 검색능력은 대상 데이터베이스가 제공하는 검색기능에 국한될 뿐 이를 바꿀 수는 없다. 예를 들어, 3개 용어를 불리언 연산자로 조합하여 검색하도록 하는 기능을 대상 데이터베이스가 지원하지 못하는 경우, 통합검색솔루션이 이를 가능하도록 할 수는 없다. 통합검색솔루션은 대상 데이터베이스의 검색기능성을 향상시킬 수는 없고 단지 그 기능들을 활용할 뿐이다.

## 결언

위에 지적한 몇 가지 제한점에도 불구하고 통합검색솔루션은 국내외를 막론하고 널리 보급되고 있다. 국내에도 외국제품이 생각보다 여러 종류가 도입되어 있다. 국내제품과 외국제품은 비교항목에 따라 서로가 비교우위를 가진다. 일단 가격 면에서 외국제품은 리스(lease)의 형태를 가져 초기 도입비용이 국내제품과 유사하나 이들 제품의 매년 구독료는 일반 국내제품의 유지보수비용보다 훨씬 높으며 사용하는 동안에는 꼼짝없이 해당 금액을 지불해야 한다. 이는 국내 제품에도 우리나라 기관이 준수해야 할 “원칙”이다. 대상 사이트 검색에서는 만약 도입 도서관이 주로 외국 사이트를 검색한다면 외국제품의 우세를 부인할 수 없다. 하지만 국내 사이트 검색이 많다면 외국제품의 검색방식은 n-gram 방식이며 이에 대한 부작용은 이전에 설명한 바 있다. 또한 국내 사이트는 검색페이지를 자주 재구성하기 때문에 검색결과화면을 단일화하지 않고 대상 사이트의 검색화면을 그대로 사용하는 것을 고려해 볼 수 있다. 이는 대상 기관의 재구성으로 인한 오류가 전체 통합검색시스템에 미치는 오류를 부분적으로 제한할 수 있는 장점이 있다.

통합검색시스템은 아주 편리한 도구이지만 그것에 대해 우리가 무엇을 기대할 수 있고 그렇지 않은가에 대한 명확한 기대치를 가질 필요가 있다. 어떤 시스템이 모든 문제를 해결해 줄 것이라고 기대하기보다, 그것에 대해 공부하고, 관리자인 동시에 최종이용자의 입장으로 장단점을 분석하여 개선을 요구하며, 적어도 휘둘리지는 말자는 것이다. ☹

7) 여기서 구매와 구독은 도입형태를 의미하는 것이 아니라 전자는 한번 도입하면 지속적으로 특정 수준을 유지하는 것을 가정하였고, 구독은 지속적인 서비스를 받아야 한다는 의미로 사용되었다.