Journal of The Korea Society of Computer and Information
Vol. 24 No. 7, pp. 39-44, July 2019
www.ksci.re.kr
https://doi.org/10.9708/jksci.2019.24.07.039

Performing Data Integration: Handed-code Approach vs. Tool-based Approach

Heung-Seo Koo*

Abstract

Data integration technology is one of the key elements in building data warehouses or big data, and is used to combine data from multiple sources and provide an integrated view to users. Traditionally, the performance of data integration uses a handed-code approach or a tool-based approach that utilizes data integration tools such as ETL. There is a debate about which methods are efficient. This study is conducted to give practitioners preparing for a data integration project an insight into how to perform data integration. This paper examines the views of experts on the controversy over the adoption of ETL tools that have been on the agenda of the data integration area for over a decade.

▸Keyword: Data Integration, ETL, Handed-code Approach, Tool-based Approach, Big Data

I. Introduction

오늘날 정보시스템은 거의 대부분의 기관에서 비즈니스 활동의 근간을 이루는 중요한 인프라다. 단일 질의 인터페이스 하에서 정보 사일로(information silos)라고 불리는 이기종 데이터 소스를 병합하는 문제는 오랫동안 존재해 왔다. 데이터 통합(data integration) 기술은 데이터웨어하우스(data warehouse) 또는 빅데이터를 구축하는데 중요한 요소 중 하나이며, 여러 소스(source)에 있는 데이터를 결합하고 사용자에게 통합 뷰를 제공하기 위해 사용한다. 데이터 통합 프로세스는 기업 합병의 결과로 두 회사의 데이터베이스를 병합해야 하는 경우에서부터 서로 다른 생물정보 저장소의 데이터를 결합하는 경우와 같이 다양한 상황에서 수행한다. 사용자가 범죄통계, 날씨, 호텔, 인구통계 등 도시에 대한 다양한 정보를 질의할 수 있는 웹애플리케이션을 요구한다고 가정하자. 전통적으로 정보는 단일 스키마로 단일 데이터베이스에 저장하므로, 이러한 폭넓은 정보를 제공하려면 산재되어 있는 여러 소스 시스템에서 데이터를 수집해야 한다. 통합 뷰를 제공할 수 있는 다양한 데이터를 수집할 수 있는 자원들이 있더라도 기존 범죄 데이터베이스, 날씨 데이터베이스, 인구통계 데이터베이스에 있는 데이터가 중복될 수 있으므로 선별적으로 추출해서 목표 시스템의 데이터 형식에 맞게 변환해서 저장해야 한다[1].

ETL(Extract-Transform-Load)은 여러 소스의 데이터를 병합하는 데 사용되는 세 단계 (추출, 변환, 적재)로 구성된 데이터 통합 방식이다. 또한 데이터웨어하우스를 구축하는데도 사용되고, 데이터 마이그레이션을 수행하는데도 사용되는 활용도가 높은 데이터 이주 방식이다. 이 프로세스 중에 소스 시스템에서 데이터를 추출하여 분석할 수 있는 형식으로 변환하여 데이터웨어하우스 또는 다른 형태의 목표 시스템에 저장한다[2].

ETL 작업의 수행은 프로그래밍 언어나 데이터베이스 언어, 또는 스크립트 코드 등을 사용하여 직접 구현한 수동코딩 방식(hand-coded approach)과 ETL 프로세스를 지원하는 소프트웨어를 활용한 도구기반 방식(tool-based approach)으로 분류된다. 국내의 경우 연구 [3,4]를 비롯하여 최근 수년 간 우리가 경험한 공공기관의 대부분 데이터 통합 프로젝트가 시스템 통합 업체에 의해 수동코딩 방식으로 수행되었다. ETL 소프트웨어는 1990년대 중반에 처음 개발되어 데이터 통합 영역에서 오랫동안 활용되어 왔다. 그럼에도 불구하고 데이터 통합의 수행이 수동코딩 방식과 ETL 도구를 활용한 도구기반 방식 중 어느 방식이 효율적인지에 대한 논쟁이 현재도 진행 중이다.

이러한 이유는 ETL 프로세스가 기본적으로 추출-변환-적재 단계로 구성되지만 데이터 프로파일링, 데이터 정제, 그리고 소

• First Author: Heung-Seo Koo, Corresponding Author: Heung-Seo Koo
*Heung-Seo Koo (hskoo@cju.ac.kr), Division of Software Convergence, Cheong-Ju University
• Received: 2019. 05. 22, Revised: 2019. 06. 13, Accepted: 2019. 06. 13.
• This work was supported by the Research Grant(2017) of Cheong-Ju Univ.

스 데이터와 목표 데이터 간에 스키마 매핑 관계 및 변환 규칙의 정와 같은 연관 작업도 포함한다. 또한 여러 하위 프로세스를 포함할 수도 있다. 데이터 통합 작업은 소스 시스템에서 목표 시스템으로 데이터를 복사하는 단순한 작업이 아니고 매우 복잡한 프로세스이다. 이러한 이유로 데이터 통합의 수행을 위해 수동코딩 방식과 도구기반 방식 간에 우수성을 정량적인 결과로 제시하기 어렵다. 연구 [5]는 12개 컬럼으로 구성된 단일 테이블에 1만개 레코드를 저장한 테스트 환경에서 수동코딩 방식과 도구기반 방식 (Talend Open Studio, Pentaho Data Integration) 간의 성능을 비교하였다. 연구 [6]은 6개 테이블에 약 2,000만개의 데이터가 저장된 소스 데이터베이스에서 약 90만개의 데이터를 추출해서 목표 데이터베이스로 이관하는 테스트를 수행하였다. Talend를 이용한 도구기반 방식과 MS-SQL Server의 T-SQL을 이용한 수동코딩 방식 간의 성능을 비교하였다. 이들 연구에서는 수동코딩 방식이 각각 14%, 21%, 약 4배 정도 성능이 우수한 것으로 나타났다. 이 연구들의 한계는 1:1 스키마 매핑구조의 단순한 ETL 프로세스에서 수행성능만 비교했다는 점과, 수동코딩 방식에 비해 도구기반 방식이 우수한 기능을 활용할 수 있다고 정성적인 결론을 냈다는 점이다.

지금까지 데이터 통합 수행을 위한 수동코딩 방식과 도구기반 방식 간의 정량적 비교연구 결과가 실용적인 수준으로 제시되지 못한 이유는 데이터 통합 환경이 매우 다양하고 복잡하며, 데이터 통합에 관한 산업계 혹은 학술적 표준이 없기 때문이다. 이런 이유로 데이터 통합 표준 프로세스가 부재하여 제품들마다 지원 기능이 다양하고 고유의 인터페이스를 지원하고 있다.

이를 위해 본 연구에서는 수동코딩 방식과 도구기반 방식에 대한 데이터 통합 실무전문가들의 견해를 살펴봄으로써 데이터 통합을 준비하는 실무기술자들에게 ETL 프로세스의 수행방식에 대한 실용적 관점을 제시하고자 한다. 세부 연구 내용은 최근 10여년 간 논쟁의 주제인 ETL의 수행 방식, 즉 수동코딩 방식과 도구를 활용한 도구기반 방식에 대한 장점과 문제점을 분석하고 관련 전문가들의 견해를 살펴봄으로서 수행 방식의 올바른 선택에 대한 통찰력을 얻고자 한다. 또한 ETL 도구의 선정을 위한 타당한 방법이 있는지를 살펴본다. 연구방법은 데이터 통합 영역에서 오랫동안 진행되어 온 ETL 도구에 대한 논쟁에 관해 2006년부터 2018년까지 걸친 관련 문헌 [7,8,9,10,11,12,13,14,15]들을 시간 순으로 살펴봄으로서 ETL 소프트웨어 기술발전에 따른 실무전문가들의 견해를 살펴본다. 검토한 문헌의 유형은 전문가 블로그, 기업백서, Gartner의 시장조사 보고서를 기반으로 한 전문가 블로그, 데이터 통합 전문가 인터뷰, 전문가 토론 웹사이트 등이다.

국내의 많은 통합 정보시스템 구축 프로젝트에서 데이터 통합의 수행 방식으로 수동코딩 방식을 주로 사용하는 상황에서 이 연구결과가 데이터 통합을 준비하는 실무자들의 의사결정에 도움을 줄 수 있을 것으로 기대한다.

본 논문의 구성은 2장에서는 먼저 관련 배경지식으로 데이터 통합에 대해 살펴보고, 3장에서는 데이터 통합의 수행방식에 대한 전문가 견해들을 살펴본다. 4장에서는 도구의 선정방법에 대해 살펴보고, 마지막으로 5장에서 시사점을 제시하고 결론을 맺는다.

II. Related Work

1. Overview of Data Integration

데이터 통합은 데이터를 가치 있는 비즈니스 통찰력으로 전환하기 위해 이질적인 소스 시스템으로부터 서로 다른 데이터를 병합하는데 사용되는 기술 및 비즈니스 프로세스이다. ETL (추출, 변환, 적재) 기술은 실제 데이터 통합의 가장 일반적인 형태이지만, 복제(replication)나 가상화 (virtualization)를 비롯한 다른 기술도 활용한다[16]. ETL은 1990년대 데이터웨어하우스와 함께 발전했다. 전통적인 ETL 프로세스는 산재되어 있는 여러 소스에서 데이터를 추출하고, 일관된 형식으로 변환하여, 데이터웨어하우스에 맞춤식(customization)으로 삽입하는 프로세스를 지칭하며[17], 이러한 기능을 지원하는 소프트웨어를 ETL 도구라고 한다. ETL 프로세스는 주기적으로 소스 시스템에 해당하는 OLTP 데이터베이스 또는 비OLTP 시스템에서 데이터를 추출하고 데이터웨어하우스 스키마에 맞게 변환하여 데이터웨어하우스에 적재하는 형태가 전형적이다.

ETL은 비용이 많이 들고 노동 집약적이며 미션 크리티컬(mission critical)하고 데이터웨어하우징 프로젝트의 성공에 중요한 요소이기 때문에 ETL은 데이터 통합 작업에 매우 중요하다. 어려움은 데이터 통합과 데이터 정제 작업의 결합에 있다. 일반적인 통합 문제로서, 데이터가 특정 데이터 저장소에서 다른 데이터 저장소로 이동 또는 전송되는 경우 스키마 및 값 매핑의 세부사항이 중요한 역할을 한다. 동시에 데이터는 매우 불규칙한 상태(중복, 제약 및 비즈니스 규칙 위반, 필드의 내부 구조의 불규칙성 측면 모두) 일 수 있다. 그러므로 비구조적인 데이터를 전형적인 관계형 구조로 변환하는 효과적인 방법을 찾는 것은 매우 어렵다. 또한 ETL 프로세스는 오류를 복원 할 수 있는 방식으로 표준화, 최적화 및 실행하기가 어렵다[18].

ETL 및 데이터 정제 과정은 데이터웨어하우스 예산 중 1/3 이상을 소비하는 것으로 추산되며 이는 데이터웨어하우스 프로젝트의 개발기간의 80%까지 추가로 증가할 수 있다고 추정되고 있다[19]. 최근의 확장 ETL 도구들은 데이터 통합을 지원하는 기능을 포함하면서 데이터 통합을 위한 주요 범용 도구로 활용되고 있다. 또한 ETL 소프트웨어는 한 데이터베이스의 데이터를 새로운 데이터베이스로 복제하는 데이터 이관 도구로도 많이 사용되고 있다. 많은 기관에서 ETL 도구의 자체 개발을 선호하는데, 그 이유는 이러한 도구의 도입비용이 데이터웨어하우스 운영시 총비용의 55%를 차지하고 두 번째로 이러한 도구의 학습비용이 높기 때문이다.

2. Limitations of ETL Tools

ETL 소프트웨어는 1990년대 중반에 처음 개발된 이후 지속적으로 기술이 발전하여 크게 기능이 개선되었지만, 비즈니스 인텔리전스의 요구도 복잡해지고 있어 여전히 많은 한계를 가지고 있다. 여러 연구에서 ETL 도구의 장점, 단점, 한계점을 제시하였으며, 다음은 [19]에서 제기한 ETL 도구의 한계점 중 주요 내용이다.

- 많은 ETL 도구들이 기업 독자적인 API(Application Programming Interface) 및 메타데이터 형식(meta data format)을 사용하여 상호연동성 미흡으로 인해 ETL 도구들을 연계하기 어렵다.
- 수동코딩 방식과 도구기반 방식의 ETL 프로세스 각각의 강점, 약점이 Zode에 의해 연구되었다. 각 방식은 장단점이 있으며 적합한 방식을 선택하기 위한 요소들이 제시되었다. 그러나 ETL 도구의 선택은 여전히 연구주제로 남아 있으며, 각 기관에 적합한 평가기준을 설정하는 일이 어려운 과제이다.
- 현세대의 ETL 도구들이 비즈니스 요구사항을 체계적으로 캡처링하여, 이들을 정확함과 품질 요구사항을 충족하는 최적화된 설계로 전환하는 기능을 거의 지원하지 않는다.
- 차세대 BI 솔루션들은 더 많은 사용자 요구사항을 해결하도록 요구받고 있다. 예를 들면 실시간에 가까운 실행, 구조 및 비구조 데이터의 통합, 운영 애플리케이션과 분석 애플리케이션 간에 좀 더 유연한 데이터 흐름 등이 있다. 이러한 요구사항을 해결하려면 데이터 통합 흐름설계의 복잡도가 증가하는데 이에 대한 기능 지원이 미흡하다.

III. Approaches of Performing Data Integration

실무전문가들 사이에는 데이터 통합의 수행을 수동코딩 방식으로 할 것인지와 도구기반 방식으로 할 것인지에 대해 오랫동안 논쟁의 주제였다. 이번 장에서는 수동코딩 방식과 도구기반 방식에 관한 실무 전문가들의 견해를 살펴본다.

본 연구에서는 1990년대 중반에 처음 개발되어 데이터 통합 영역에서 오랫동안 활용되어 온 ETL 도구의 활용에 대한 논쟁에 관해 2006년부터 최근까지 10여년에 걸친 관련 문헌[7,8,9,10,11,12,13,14,15]들을 살펴봄으로서 데이터 통합의 수행 방식에 대한 통찰력을 얻고자 한다. 검토한 문헌의 유형은 전문가 블로그, 기업백서, Gartner의 시장조사 보고서를 기반으로 한 전문가 블로그, 데이터 통합 전문가 인터뷰, 전문가 토론 웹사이트 등이다.

데이터 통합 영역에서 수동코딩 방식과 도구기반 방식의 사용에 대한 논쟁은 2006년 David Aldridge[7]가 전문가 커뮤니티 블로그에 처음 견해를 밝힘으로써 촉발되어 최근까지 10여 년간 이어지고 있다. David는 도구기반 방식의 문제점을 13가지 제시하였다. 주요 내용은 다음과 같다.

- 외부 ETL 개발자나 컨설턴트들이 프로젝트 초기 시점에 소스 데이터에 대한 이해도가 낮다.
- ETL 개발자들이 데이터베이스 전문기술을 거의 활용하지 않아 수행성능 효율이 떨어진다.
- ETL 도구들이 데이터베이스 전문기술을 활용하지 않아 수행성능 측면의 효율성이 떨어지고, 대용량 데이터의 경우 처리시간이 오래 걸린다.
- ETL 도구들이 데이터베이스에 저장된 성능관련 메타정보의 활용 능력이 부족하다.
- ETL 도구들이 트랜잭션을 확장할 수 없어 복잡한 트랜잭션의 제어가 어렵다.
- ETL 도구의 구입비용과 컨설팅 비용이 높다.

2007년에 Madhu[8]는 ETL 도구들이 1990년 중반에 처음 개발된 이후 기술적으로 발전하여 ETL 도구들의 기능이 크게 개선되었다고 주장하였다. 2012년에 Craig[9]는 데이터 통합 프로젝트에서 중소기업의 경우 수동코딩 방식이 지배적이며, 대기업의 경우도 ETL 도구를 사용하고 있지만 많은 부분을 수동코딩 방식을 사용하고 있다고 밝혔다. 이러한 이유는 내부 인력에게 익숙한 기술이라는 점과 ETL 도구가 너무 고가라는 인식 때문에 수동코딩 방식에 의존한다고 하였다. 여전히 많은 기업에서 데이터 통합을 위해 수동코딩 방식을 사용하고 있으나, 도구를 활용하여 데이터 통합 프로세스를 자동화 할 때가 된 것 같다는 의견을 냈다. 2013년에 Gary Nissen[10]은 데이터 통합 프로젝트는 수동코딩 방식과 도구기반 방식 중 선택의 문제가 아닌, 엄격한 프로세스 관리 및 규정준수가 중요하다고 지적하였다. 데이터 통합 프로젝트의 범위와 요구사항을 정확하게 파악한 후, 수동코딩 방식과 도구기반 방식 중 선택해야 한다. 수동코딩 방식의 가장 큰 문제점은 빈약한 프로세스 관리와 강제적인 규정준수가 어려운 점이라고 지적하였다. 2015년에 Francisco Blanes[11]은 수행방식의 선택은 대상 프로젝트에 종속된 문제이므로 올바른 선택을 위해 프로젝트 규모와 요구사항을 검토한 후 신중하게 결정해야 한다고 의견을 제시하였다. 예산 제약이나 수동코딩 방식으로 해결해야 하는 특별한 요구사항이 없다면 도구기반 방식을 권고하였다. 2016년에 가트너 그룹의 시장조사 보고서를 기반으로 한 Ashley Stirup[12]은 수동코딩 방식과 도구기반 방식에 대한 상충관계를 평가할 수 있는 질문 목록을 제시하였다. 2017년에 Slawomir Chodnicki[13]는 도구가 사용자 요구사항을 해결할 수 없는 경우 친숙한 프로그래밍 환경과 언어를 사용해야 하지만, 기본적으로는 도구기반 방식을 추천하였다. 2018년에 Swatee Chand[14]는 미국 지식공유 웹사이트 쿠오라(Quora)에서 ETL 프로세스는 길고 복잡한데, 이러한 다단계 작업을 수동코딩하고 디버깅하는 것은 매우 어려운 일이라고 지적하였다. Jung Tong[15]은 David Aldridge의 견해를 많은 점에서 동의하지만, 문제를 제기한 이후 10년 동안 ETL 도구들이 기술 측면과 성능

측면에서 크게 향상되어 많은 문제가 해결되었다고 주장하였다.

지금까지 살펴본 문헌들로부터 다음의 시사점을 얻을 수 있다.

- 수행 방식의 문제보다 중요한 것은 메타데이터 생성 및 관리 및 공유, 문서화 등 엄격한 프로세스 관리와 규정 준수이다.
- 현재에도 수동코딩 방식과 도구기반 방식 중 선택은 전문가들 사이에도 견해도 일치하지 않지만, 최근 10여 년 동안 주요 ETL 제품의 기능 및 성능이 많이 향상되어 정교한 제품군(suite)으로 발전하였다.
- 단기 · 장기 비용을 모두 고려한 프로젝트의 총소유비용(TCO: Total Cost Ownership)을 고려하여 결정해야 한다. 단기적으로는 수동코딩 방식이 비용이 적게 들지만 느린 개발속도와 유지관리 비용을 고려하면 장기적 비용은 증가한다. 문헌 [12]는 가트너그룹의 보고서를 인용하여 수동코딩 방식은 구현 비용을 20% 감소할 수 있지만, 유지관리 비용이 200% 증가할 수 있다고 하였다.
- 수동코딩 방식은 특정 상황에서만 적합하다. 유지관리가 많이 필요하지 않은 목표가 매우 명확하고 간단한 프로젝트에서 적합하다. 또한 사용자 요구를 충족할 수 있는 도구가 없는 경우에도 시도할 수 있다.
- 수동코딩 방식에 의존하는 주된 이유 중 하나가 ETL 도구가 너무 고가라는 인식 때문이므로, 예산 범위 내의 도구를 선정하라. 고가의 도구가 기술적 경험 부족을 보완해 주지는 않는다. 도구들의 가격대와 복잡도 수준이 다양하므로 여러 가격대의 도구를 검토하는 것이 필요하다. 사용자 요구를 해결하는데 낮은 가격대의 도구가 적합한지 좀 더 고가의 도구가 적합한지를 검토해야 한다. 문제는 이러한 도구들의 비교 검토가 어렵다는 점이다.

IV. Selection Methods of ETL Tools

데이터 통합의 효율적인 수행에 절대적 영향을 미치므로 데이터 통합에 적합한 도구를 선정하는 작업은 매우 중요하다. 데이터 통합을 수행하려면 데이터 프로파일링, 데이터 품질검사 등 여러 단계에서 다양한 도구가 필요하지만 데이터 추출, 변환, 적재가 중심 작업이므로 ETL 도구의 선정방법에 관한 문헌들을 통해서 데이터 통합 도구의 선정방법에 대한 타당한 방법이 있는지를 살펴본다.

Scott[20]은 ETL 도구의 선택 기준, 적용 시나리오, 정량적 평가방법, 평가절차로 구성된 표준 테스트 세트 (suite)를 제시하였다. 6가지의 성능지표와 8가지 유형의 평가기준을 광범위하고 구체적으로 제시하고 평가절차까지 개발하였다. 그러나 이 방법은 ETL 소프트웨어의 특성들을 여러 하위 평가요소를 포함하는 평가유형에 따라 세분화하고 각각의 평가기준에 대한 가중치를 설정하는 작업이 복잡하기 때문에 실제 적용하기 어렵다. 이 방법이 대중화되려면 다양한 적용사례가 개발되어 적용방법을 구체화 및 단순화하는 것이 필요하다.

Zodes[8]는 ETL 도구를 선정하기 위한 12가지 고려사항을 제시하였다. 이들 고려사항은 플랫폼 독립성, 소스 타입의 독립성, 메타데이터 지원, 지원 기능, 사용의 용이성, 버전제어, 병렬처리 기능, 스타스키마 지원, 디버깅 기능, 스케줄링 기능, 배치(deploy) 기능 , 재사용성이다. Aman[19]은 ETL 도구를 선정하기 위한 10가지 평가요소로 구성된 프레임워크를 제시하였다. 이는 ETL 도구 개발 시에도 벤치마크 도구로 사용할 수 있다. 제시된 평가요소는 멀티쓰레드 처리 등과 같은 기본 기능의 지원을 포함하여 GUI(Graphical User Interface) 지원, 점진적 갱신 지원, 공통작업의 통합, 공통기능의 통합, 레거시 데이터 지원, 데이터웨어하우스를 위한 실시간 클릭스트림 지원, e-비즈니스 환경 지원, 플랫폼 독립성과 확장성, 비즈니스 규칙과 API에 대한 확인 기능이다. Zodes와 Aman이 제시한 방법은 고려사항 또는 평가요소의 항목들이 포괄적이어서 평가대상 도구들의 정확한 평가나 도구들 간의 차별성을 식별하기 쉽지 않다.

앞에서 살펴본 것처럼 도구 선정은 쉬운 작업이 아니며 효과적인 도구의 평가기준의 개발은 여전히 연구주제로 남아 있다. 이러한 이유는 데이터 통합이 여러 하위 프로세스로 구성될 수 있는 복잡한 프로세스이므로 ETL 도구도 복잡하게 구성되어 있다. 좀 더 근본적인 문제는 데이터 통합에 대한 산업계 또는 학술적 표준이 없기 때문에 제품들마다 지원 기능이 매우 다양하고 복잡하여 제품들 간의 기능을 비교하기 어렵다.

그럼에도 불구하고 도구 선정이 필요한 경우, 먼저 데이터 통합에 대한 요구사항을 분석하고, 이를 바탕으로 Zode와 Aman의 고려사항, 평가방법, 그리고 Ashley의 질문목록을 참고하여 기업 내부의 도구 선정기준을 개발하는 것이 바람직하다. 이때 평가요소의 우선 순위와 각 요소별 세부 수준을 설정하는 것도 필요하다.

적합한 도구를 선정하는 작업은 매우 중요하다. 그 이유는 데이터 통합과 ETL 영역에는 산업계 또는 학술적 표준이 없기 때문에 도구들 간의 호환성이 매우 낮다. 한 ETL 제품에서 다른 ETL 제품으로 전환하는 것이 매우 어려워, 한 ETL 제품에서 개발된 코드들을 다른 ETL 제품으로 이관하거나 변환할 수 없다. 그러므로 ETL 도구의 선정은 매우 신중하게 이루어져야 한다.

V. Conclusions

다양한 데이터에 대한 통합적 조회와 통합적 데이터 분석에 대한 요구가 증가함에 따라 데이터 통합에 대한 요구가 점차 증가하고 있다. 통상적으로 데이터 통합의 수행은 수동코딩 방식이나 ETL 소프트웨어와 같은 데이터 통합 도구를 활용한 도구기반 방식을 사용하는데, 어느 방식이 효율적인지에 대한 논쟁이 지난 10여 년간 진행되어 왔다. 본 연구는 데이터 통합을

준비하는 실무자들에게 데이터 통합의 수행방식에 대한 실용적 관점을 제시하기 위해 수행하였다. 또한 데이터 통합 도구의 선정을 위한 타당한 방법이 있는지도 살펴보았다.

이번 연구의 결과는 다음과 같은 시사점을 준다.

첫째, 단기 · 장기 비용을 모두 고려한 프로젝트의 총소유비용을 고려하여 결정해야 한다. 단기적으로는 수동코딩 방식이 비용은 적게 들지만 느린 개발속도와 유지관리 비용을 고려한 장기적 비용은 증가하기 때문이다.

둘째, 현재에도 수동코딩 방식과 도구기반 방식 중 선택은 전문가들 사이에도 견해도 일치하지 않지만, 특별한 요구사항이 없는 경우는 도구기반 방식이 바람직하다. 유지관리가 많이 필요하지 않은 목표가 매우 명확하고 간단한 프로젝트 이외의 경우는 도구기반 방식이 효율적이다.

셋째, 고가의 도구가 기술적 경험 부족을 보완해 주지는 않는다. 도구들의 가격대와 복잡도 수준이 다양하므로 여러 가격대의 도구를 검토하는 것이 필요하다. 사용자 요구를 해결하는데 낮은 가격대의 도구가 적합한지 좀 더 고가의 도구가 적합한지를 검토해야 한다.

넷째, 외부의 도구 개발자나 컨설턴트들이 프로젝트 초기에 소스 데이터에 대한 이해도가 낮은 문제를 해결할 수 있는 방안을 모색해야 한다. 소스 데이터의 정확한 파악이 프로젝트 규모에 대한 올바른 판단이 가능하다.

여섯째, 데이터베이스 지식을 갖춘 해당 ETL 도구에 대해 많은 경험을 가진 개발자를 확보해야 한다.

데이터 통합을 준비하는 실무자들이 고려해야 할 여러 문제가 있다. ETL 도구를 선정하기 위한 평가기준의 개발이 여전히 연구 영역에 남아 있어서 적합한 도구 선정이 어렵다는 점이다. 그리고 위에서 제시한 시사점들 중 도구기반 개발자가 많은 경험을 가졌는지 여부 확인 문제, 그리고 외부 개발자가 프로젝트 초기에 소스 데이터에 대한 이해도를 높이는 문제는 현재 통합정보시스템 구축 및 운영을 아웃소싱하는 공공기관에서는 현실적으로 실현하기 쉽지 않다. 이에 대한 해결책은 단기적으로는 데이터 통합 외부 컨설턴트에 의존하지면 장기적으로는 내부 인력을 육성해야 한다. 시스템 통합 업체 또는 도구 공급업체에 일임해서는 안되고, 기관 내부에서 데이터 통합에 관련된 최소한의 전문 지식을 축적해야 한다. 도구에 대한 지식과 데이터 통합 프로세스에 대한 이해도가 낮으면 적합한 도구의 선정과 적절한 통합 프로세스의 감독을 수행할 수 없기 때문이다.

REFERENCES

[1] Data Integration, *https://en.wikipedia.org/wiki/Data_integration*, last edited on 2019.05.01.

[2] ETL: What it is and why it matters, *https://www.sas.com/en_us/insights/data-management/what-is-etl.html*, accessed on 2019.03.10.

[3] Sung-Ho Shin, Min-Ho Lee, et. al., "A Data Migration Model and Case Study for Building Management System of Science and Technology Contents", Journal of the Korea Society of Computer and Information, Vol.16, No.11, Nov., 2011.

[4] Hee-Seo Park, Hee-Chern Kim, "A Data Migration Method for Developing GIS-based Fisheries Resources Information Systems", Journal of Digital Convergence, Vol.11, No.6, Jun. 2013.

[5] Jong-Keun Choi, Assessment of Open Source ETL Tools for Data Migration, Soongsil Univ. Master's Thesis, 2011.6.

[6] Sang-Hyun Park, A Study on Performance Increase of ETL Data Migration, Soongsil Univ. Master's Thesis, 2015.6.

[7] David Aldridge, A List: Ten Reasons Not To Use An External ETL Tool, Blog in The Oracle Sponge, *https://oraclesponge.wordpress.com/2006/12/20/a-list-ten-reasons-not-to-use-an-external-etl-tool/*, 2006.12.20.

[8] Madhu Zode, The Evolution of ETL: From Hand-coded ETL to Tool-based ETL, White Paper, Cognizant Technology Solutions, 2007.

[9] Craig Stedman, Automated data integration tools versus manual coding, Decision time, SearchDataManagement.com, 2012.08.12.

[10] Gary Nissen, Is Hand-Coded ETL the Way to Go? Absolutely Yes, or Absolutely No, Depending, Article, *http://www.garynissen.com/etl-hand-code-or-tool/*, 2013.02.20.

[11] Francisco Blanes Martin-Portugues, ETL tools vs Hand-coded ETL, *https://www.linkedin.com/*, 2015.07.31.

[12] Ashley Stirrup, Hand Coding vs. Tools: Our Take on Gartner's Report, Blog in Talend, *https://www.talend.com/*, 2016.10.12.

[13] Slawomir Chodnicki, To code, to ETL, or to SQL?, blog in Twineworks, Twineworks, *https://blog.twine works.com/are-we-doing-data-pipelines-wrong-4176b5f7964f*, 2017.12.05.

[14] Swatee Chand, What are the advantages of using an ETL tool vs. handed coding? And is Talend a good ETL tool?, Quora.com, 2018.02.02.

[15] June Tong, ETL Tools vs. Hand Written SQL, White Paper, Sesame Software, accessed in 2018.8.16.

[16] Timothy King, Data Integration vs. Data Migration: What's the difference?, Solutions Review, *https:// solutionsreview.com/data-integration-vs-data-migration-what.h*

tml, accessed on 2018.08.16.
[17] Margaret Rouse, Data Migration, Blog in SearchCIO, posted 2016.10.14.
[18] Panos Vassiliadis, Alkis Simitsis, “Extraction, Transformation, and Loading”, Encyclopedia of Database Systems, 2009.
[19] Aman Partap Singh Pall, Jaiteg Singh, “ETL Methodologies, Limitations and Framework for The Selection and Development of an ETL Tool”, International Journal of Research in Engineering and Applied Sciences, Vol.6, No.5, 2016.05.
[20] Scott Henry, et al., “Engineering Trade Study: Extracrt, Transform, Load Tools for Data Migration”, Proceedings of the 2005 Systems and Information Engineering Design Symposium, IEEE, pp.1~8, 2005.

Authors

Heung-Seo Koo received the B.S., M.S. and Ph.D. degrees in Computer Science from Inha University, Korea, in 1985, 1989 and 1993, respectively. Dr. Koo is a Professor in the Division of Software Convergence, Cheong-Ju University, Cheongju-shi, Korea, since 1994. He is interested in data modeling, data quality, data integration, ETL, and big data infrastructure architecture.