

ESB 상에서 데이터 품질 관리를 위한 서비스 개발

문은영⁰, 최병주, 이정원
이화여자대학교 컴퓨터학과
(mney, bjchoi, jungwony)@ewha.ac.kr

Developing Service for Managing Data Quality on ESB

Eunyoung Moon⁰, Byoungju Choi, Jung-Won Lee
Dept. of Computer Science & Engineering, Ewha Womans University

요 약

기존의 컴포넌트 기반으로 응용 프로그램을 개발하던 소프트웨어 개발 방법론이 분산 컴퓨팅 환경에서 대규모 시스템 통합의 요구가 생겨남에 따라 서비스-기반 구조(SOA: Service-Oriented Architecture)로 점차 변화하고 있다. SOA를 구현하기 위한 최신 통합 기술인 엔터프라이즈 서비스 버스(ESB: Enterprise Service Bus) 상에서 상호 작용하는 시스템 간에 교환되는 데이터 품질을 보장할 수 있다면 더 나은 서비스를 제공할 수 있다. 본 논문은 기존에 개발된 DAQUM(Data Quality Measuring tool)[1]을 이용하여 ESB 상에서 통합된 시스템들간의 상호 작용하는 데이터 품질을 보장하기 위한 서비스를 개발한다.

1. 서론

점차 시스템의 규모가 확장되고, 비즈니스 로직이 복잡해짐에 따라, 모든 작업을 메인 컴퓨터에 맡기는 중앙 집중형 컴퓨팅 시스템 환경에서 분산 컴퓨팅 시스템 환경으로 변화해 왔다. 서버-클라이언트의 기본 개념을 가지는 분산 컴퓨팅은 CORBA, DCOM과 RMI의 프로토콜을 사용하여 컴포넌트 기반의 분산 컴퓨팅 환경을 구축하였다. 컴포넌트 기반의 분산 컴퓨팅 환경은 서버와 클라이언트와의 연결이 중요하기 때문에 연결 지향적 시스템(Connection-Oriented System)으로도 불린다. 그러나 연결 지향적인 시스템의 경우에는 네트워크의 상태에 따라 메모리나 연결의 손실 등 귀중한 자원의 낭비가 발생된다. 또한 비즈니스 규모가 커지고 기업간의 전자 상거래가 활발해짐에 따라 시스템과 시스템을 통합시키는 '웹 서비스'와 같은 새로운 분산 컴퓨팅 통합기술이 시작되었다[2]. 더 나아가 이러한 서비스들은 특정 프로토콜에 의해 제공되고 인터넷 상에서 상호 작용하기 때문에 엔터프라이즈 수준의 시스템들을 통합하고 협업 가능하게 하기 위해서는 특별한 분산 컴퓨팅 구조를 지원할 수 있는 서비스 개발을 위해 서비스-기반 구조(SOA: Service-Oriented Architecture)로 진화하고 있다[3].

SOA에서 서비스로 구성된 시스템의 경우 각 서비스들은 다른 서비스에 독립적으로 작업을 수행할 수 있으며, 서로의 인터페이스를 통해서만 연결된다. 시스템을 구성한 후, 일반적으로는 시스템을 시뮬레이션 함으로써 각 서비스들이 잘 연결되었는지 확인하지만 이는 서비스 인터페이스에 국한된 부분으로 상호 작용하는 실제 데이터의 올바른 사용은 확인할 수 없다. 그러나 서비스들이 정확하게 작업을 수행하는 지를 검사하기 위해서는 서비스 간에 흐르는 실제 데이터의 품질을 측정할 필요가 있다. 본 논문에서는 SOA를 지원하기 위한 최신 구현 기술인 엔터프라이즈 서비스 버스(ESB: Enterprise Service Bus) 상에서 서비스들이 통합되는 과정에서 실제 데이터에 대한 품질을 보장해주는 DAQUM(Data Quality Measuring tool) 서비스를 개발한다. 개발된 DAQUM 서비스는 SOA의 기본 원리를 지원하기 위한 ESB의 독립적인 기능으로 사용될 수 있다.

본 논문의 구성은 다음과 같다. 2장에서는 관련 연구로 ESB와 DAQUM에 대해 소개하고, 3장에서는 ESB 상에서의 서비스 개발 및 DAQUM 서비스의 특징과 구조, 적용에 대해 기술하고 마지막으로 4장에서 결론을 맺는다.

2. 관련 연구

2.1 ESB

CORBA와 DCOM, .NET과 같은 컴포넌트 기반의 분산 컴퓨팅의 경우 클라이언트와 서버간의 연결을 통하여 요청을 하고, 결과를 전달 받는다. 연결 지향적인 시스템은 연결이 갑자기 끊어짐으로 발생하는 정보의 손실, 비 활동 중인 클라이언트로 인한 자원 낭비, 보안 문제, 프로토콜간의 호환이 이루어지지 않는 한계를 가진다[2]. 시스템들을 연동해야 할 필요가 생겨남에 따라서 다른 시스템과 통합할 수 있는 분산 시스템 환경이 요구되었다. 이에 시스템과 클라이언트와 지속적인 연결이 아닌 요청 후 연결을 해제하고, 플랫폼에 독립적인 XML 기반의 SOAP을 사용하며, 다른 프로토콜로 구현된 시스템 환경을 통합시킬 수 있는, 인터넷 상에서의 분산 컴퓨팅이 가능한 웹 서비스가 사용되었다.

SOA는 분산 시스템을 위한 설계 방법론이다. SOA는 서로 간의 약 결합(loose coupling)되어 다른 서비스와는 독립적으로 존재하며, 서로 상호 작용하는 서비스들로 시스템을 구성한다. SOA의 기본 원리는 구현 독립적인 인터페이스의 사용, 위치 투명성(location transparency)과 상호 운용성을 강조하는 커뮤니케이션 프로토콜의 사용, 그리고 재사용 가능한 비즈니스 기능을 추상화하여 서비스를 라이브러리처럼 사용할 수 있도록 하는 서비스 정의를 포함한다[4]. ESB는 SOA의 기본 원리를 따르면서 통합 인프라 구조를 제공하는 논리적인 구조로, 이벤트 중심의 분산 시스템 환경을 가능하도록 SOA를 지원하는 논리적인 버스의 개념이다. ESB를 위해 기본적으로 커뮤니케이션, 통합, 서비스 상호 작용과 관련된 특성들을 만족시켜야 하며 그 외에도 보안, QoS(Quality of Service), 메시지 프로세싱 등과 같은 특성을 지원한다.

그림 1은 C/C++로 구현된 레거시 어플리케이션, ERP와

같은 시스템, .NET과 J2EE 플랫폼, 그리고 웹 서비스들을 신뢰성 있고 비동기식 보안 메시지 전송인 자바 메시징 서버(JMS: Java Messaging Server)를 통하여 통합하는 ESB 구조의 한 예를 보여준다.

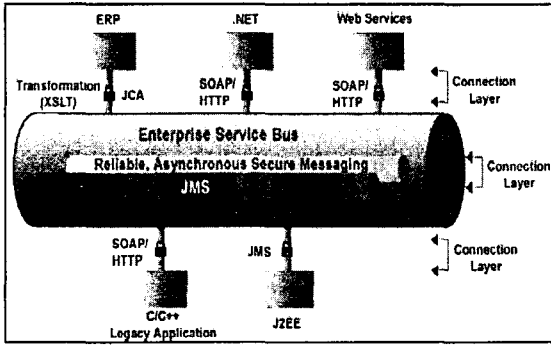


그림 1. Enterprise Service Bus 구조[5]

2.2 DAQUM

지식공학 시스템은 대용량의 소스 데이터에서 의미 있는 지식을 추출해내기 때문에 소스 데이터의 품질을 보장해야 한다. 본 논문에서 사용할 데이터 품질 관리 도구인 DAQUM은 데이터에 품질 관리 방법론인 TQM(Total Quality Management)를 적용한 TDQM(Total Data Quality Management)의 생명 주기를 이용하여 실제 데이터를 사용하는 사용자가 지정한 목적에 따라서 사용자의 관점에서 품질을 다르게 평가할 수 있도록 하면서 데이터 품질 측정, 오류 데이터 탐지, 정제하는 도구이다[1].

다음의 그림2는 DAQUM이 수행하는 업무에 대한 프로세스이다. 품질을 보장하고자 하는 소스 데이터가 저장된 데이터 베이스와 연결하고, 사용자 관점의 품질 평가를 위하여 속성을 설정하고, 데이터의 품질을 측정하여 오류데이터를 탐지하여 사용자로 하여금 각각의 오류 데이터를 정제할 수 있도록 해준다. 이때 사용자의 속성 설정 외에 오류 데이터를 탐지하는 기준은 오류데이터를 특성에 따라 분류한 오류 데이터 분류표[6]에 따라 측정하게 된다.

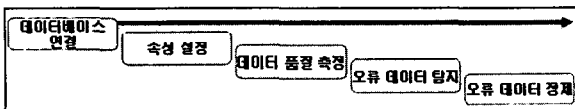


그림 2. DAQUM 프로세스

그러나 현재 DAQUM은 하나의 오프라인 데이터베이스 내에서의 데이터를 대상으로 국한되어 있다. 따라서 본 논문에서는 ESB 상에서 DAQUM을 서비스화 함으로써, SOA를 지원하는 대표적인 데이터 품질 측정 서비스로 확장하고자 한다.

3. ESB 상에서의 DAQUM 서비스화

3.1 ESB상에서의 서비스 개발 절차

SOA를 구현하기 위해 ESB 상에서 시스템을 구성 및 통합하기 위해서는 다음의 4 단계에 따라 이루어진다.

- 시스템 분석: 요구된 시스템을 분석하여, 필요한 서비스를 명세화하고 시스템 구조를 구축한다.
- 서비스 개발: 분석을 통해 명세화된 서비스를 생성하고 서비스 라이브러리에 등록한다.

- 시스템 통합 및 사용: 시스템에서 사용되는 서비스의 특성을 설정하고, 각 서비스 간의 인터페이스를 확인하고 구조를 기반으로 시스템에 필요한 서비스들을 조립한다.
- 시뮬레이션: 서비스들로 구축된 시스템이 제대로 되었는지 테스트한다.

위의 단계에 따라 서비스를 개발하여, SOA를 지원하는 시스템에 ESB 상에서 구축할 수 있다. ESB는 시스템에서 필요로 하는 서비스를 추출하여 명세화하고, 이에 따라서 기존의 서비스를 재사용하거나 새롭게 서비스를 생성하여 서비스 라이브러리에 등록하여 저장된 서비스를 사용하여 전체 시스템을 구성할 수 있는 개발 환경을 제공한다. 그림 3은 ESB 상에서 서비스를 개발, 등록, 그리고 사용하는 절차를 보여준다.

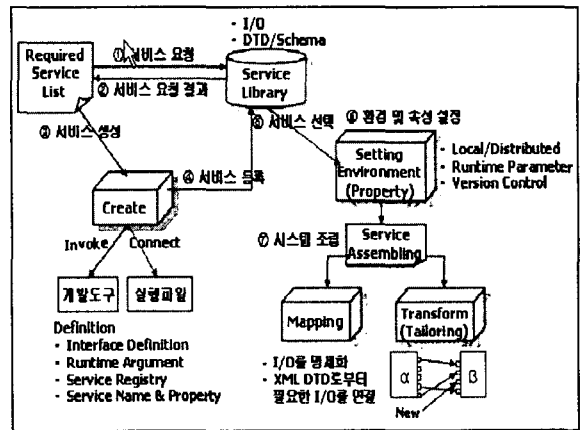


그림 3. ESB 상에서의 서비스 개발

- ① 서비스 요청: 시스템 분석을 통해 필요한 서비스 목록을 얻고 이를 필요한 서비스 라이브러리에 요청한다.
- ② 서비스 요청 결과 전송: 서비스 라이브러리에 요청된 서비스의 존재 여부를 결과로 전달 받는다.
- ③ 새로운 서비스 생성: 개발 도구를 호출하거나, 실행파일을 연결함으로써 서비스를 생성한다. 이때 서비스와 관련하여 인터페이스, 실행시간 독립변수, 레지스트리 등을 정의한다.
- ④ 서비스 등록: 새로 개발된 서비스를 서비스 라이브러리에 등록한다.
- ⑤ 서비스 선택: 시스템을 위해 필요한 서비스들을 선택한다.
- ⑥ 환경 및 속성 설정: 서비스의 환경 및 속성(분산 환경, 실행시간 매개변수, 버전 제어 등)을 설정한다.
- ⑦ 시스템 조립: XML 기반으로 명세화된 I/O를 연결함으로써 시스템을 조립한다. 서비스들은 인터페이스를 단순히 매핑시키는 방법과 서비스간의 I/O를 테일러링 하는 방법으로 전체 시스템으로 조립된다.

3.2 DAQUM 서비스

기존의 DAQUM은 오프라인에서의 데이터 품질을 보장해 주기 위한 것으로, 이미 여러 소스로부터의 정보를 통합하여 데이터베이스에 저장된 원시 데이터를 대상으로 품질을 측정하였다. 다시 말하면, DAQUM은 오프라인 상에서 오류 데이터를 측정하고, 분류하여 사용자로 하여금 오류 데이터를 정제할 수 있는 환경을 제공하였다.

ESB상에서 DAQUM을 서비스화 한다는 것은 품질 측정에 대상이 되는 데이터가 오프라인에 한정된 것이 아니라 온라인 상의 데이터에 대해서도 품질을 보장하기 위한 것이다.

DAQUM 서비스는 다음 두 가지 기능을 지원할 수 있도록 개발한다.

- 온라인 및 오프라인 데이터 품질 관리
- ESB 상에서 데이터 품질 측정을 위한 대표적인 서비스로의 개발

3.3 DAQUM 서비스의 구조

DAQUM 서비스는 SOA를 기반으로 하는 ESB 상에서 시스템을 통합 및 개발하는데 있어서 데이터 품질을 보장해 주기 위한 서비스이다. 다음 그림 4는 DAQUM 서비스의 전체 모듈의 관계를 보여준다.

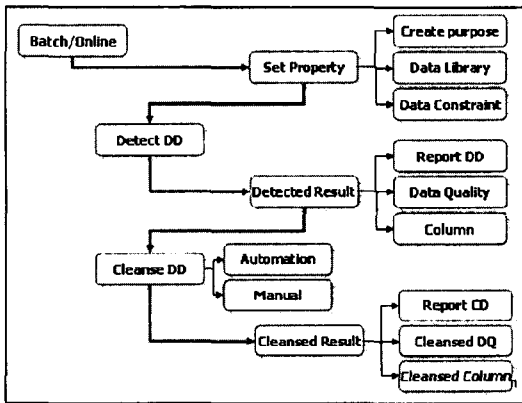


그림 4. DAQUM 서비스의 구조

DAQUM 서비스는 다음의 기술된 단계를 수행하는 모듈들로 구성되어 있다.

- 일괄처리/온라인 선택 단계(Batch/Online): DAQUM 서비스가 무엇을 대상으로 데이터 품질을 측정할 것인지를 선택하는 모듈로써 오프라인의 데이터 품질에서 온라인 데이터 품질까지 그 대상을 확장하기 위한 기능
- 속성 설정 단계(Set Property): 사용자 관점에서 데이터 품질을 측정할 수 있도록 속성들을 정의
- 오류 데이터 탐지 단계(Detect DD): 사용자에 의해 정의된 속성과 오류 데이터 분류법에 의하여 데이터 품질을 측정하여 오류 데이터를 탐지
- 오류 데이터 탐지 결과 단계(Detected Result): 분류된 오류 데이터를 리포트하고, 전체 데이터의 품질과 결립 별 데이터 품질을 그래프로 보여줌
- 오류 데이터 정제 단계(Cleanse DD): 분류 결과에 따라 사용자가 오류 데이터를 정제할 수 있는 환경을 제공
 - 자동 정제 단계(Automation): 사용자 요구에 따라 자동적으로 오류 데이터를 정제
 - 수동 정제 단계(Manual): 사용자에게 다시 정보를 입력 받아 오류 데이터를 수동적으로 정제
- 오류 데이터 정제 결과 단계(Cleansed Result): 오류 데이터가 얼마나 정제되었는지를 사용자로 하여금 쉽게 파악할 수 있도록 정제 결과를 보여줌

3.4. DAQUM 서비스의 적용

본 논문에서 개발된 DAQUM 서비스는 다음 그림 5와 같이 구성된 시스템을 구축하는데 적용할 수 있다. 적용 환경은 볼륨 거래 시스템으로 제품을 주문하는 CRM 시스템, 주문에 대한 승인과 거절을 결정하는 ERP 시스템, 그리고 거래 정보를 전달하

고 보여주는 여러 시스템들을 통합하여 전체 시스템을 구성하였다. 물품 구매자가 고객의 정보를 담당하는 CRM 시스템을 통하여 개인 정보와 함께 구매 정보를 제품 관리 시스템인 ERP에 전송한다. ERP는 전달 받은 정보를 가지고 주문을 승인할 것인지 거절할 것인지를 결정한 후, 구매자에게 e-mail을 통해서는 승인된 결과를 전송하기도 하고 대화창을 통해 거절된 결과를 전송하기도 한다. 이러한 적용 예는 여러 시스템들이 통합되고, 시스템간에 대규모 데이터가 이동하는 대표적인 시스템으로 볼 수 있다.

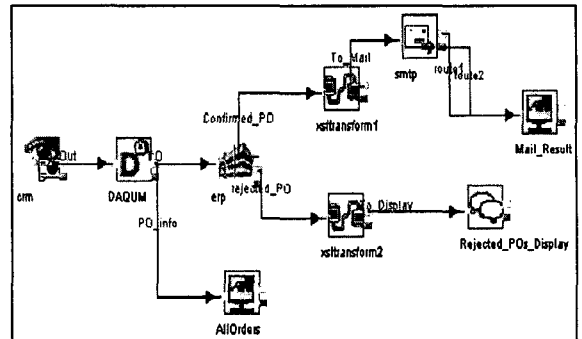


그림 5. DAQUM 서비스의 적용

여기에서, 개발된 DAQUM 서비스가 CRM과 ERP 사이에 독립적으로 사용되어 사용자가 지정한 목적에 따라서 사용자의 관점에서 데이터의 품질을 측정하여 네트워크 전송 장애로 인한 데이터 전송 오류, 구매자의 잘못된 선택 등과 같은 오류들을 실시간으로 탐지해 낼 수 있다.

4. 결론 및 향후 연구

DAQUM을 SOA를 지원하는 ESB 상에서 서비스로 개발하여, 사용자가 정의한 속성과 오류 분류법에 따라 시스템들이 상호 작용하는 데이터의 오류를 탐지하여 정제함으로써 온라인 상에서의 데이터 품질을 보장하며, 대규모 시스템간의 통합시 안전한 데이터 사용을 도모할 수 있다.

앞으로 XML을 이용한 DAQUM 인터페이스의 형식을 정의하여 자동화된 데이터 오류 탐지 및 정제 기법을 개발할 것이다.

5. 참고문헌

- [1] 김은희, 문은영, 최병주, "NET 컴포넌트 기반 데이터 정제 도구", 제 6 회 한국 소프트웨어공학 학술대회, pp252-261, 2004.2
- [2] 신민철, "기초에서 실무까지 XML 웹 서비스", FreeLec, 2004
- [3] M.P Papazoglou, D. Georgakopoulos, "Service-Oriented Computing", Communication of ACM, Vol.46 No.10, pp25-28, 2003.10
- [4] Rick Robinson, "Understand Enterprise Service Bus Scenarios and Solutions in Service-Oriented Architecture", developerWorks (<http://www-106.ibm.com/developerworks/webservices/library/ws-esbscen/>), 2004.6
- [5] Fiorano Software Inc., "Demystifying Enterprise Service Bus Technology", White Paper, pp1-9, 2003
- [6] Won Kim, Byoung-Ju Choi, Eui-Kyeoung Hong, Soo-Kyoung Kim, Doheon Lee, "A Taxonomy of Dirty Data", The Data Mining and Knowledge Discovery Journal, Vol7 No.1, pp81-99, 2003.1