EPCglobal TDT 1.0 표준을 따르는 태그 데이터 변환기의 설계와 구현

김성진° 송하주
부경대학교 전자컴퓨터정보통신공학부
manghon@manghon.com°, hajusong@pknu.ac.kr

Design and Implementation of An EPCglobal TDT 1.0 Compliant Tag Data Translator

Sungjin Kim° Haju Song
Division of Electronic, Computer, and Telecommunication Engineering
Pukyong National University

1. 서론

EPC(Electronic Product Code)[1]는 RFID 태그의 식별을 위한 코드체계이며 EPC Tag Data Standard (TDS)[2]는 기존의 코드 체계를 EPC로 표시하기 위한 규칙을 정의하고 있다. EPC 태그 데이터 변환 (Tag Data Translation, TDT))[3]은 컴퓨터로 처리하기에 용이하도록 EPC를 XML로 나타내였고 서로 상이한 태그 코드 표현 방식들 간의 변환 규칙을 정의하고 있다.

TDT v1.0에서의 변환은 동일 스킴(scheme) 내에서의 포맷 변환만을 지원하는데 각 스킴에서는 기본적으로 BINARY, TAG_ENCODING, PURE_IDENTITY, LEGACY 4가지 포맷을 지원하며, 추가적으로 ONS_HOSTNAME 포맷을 지원할 수도 있다. 그러므로 한 스킴당 16~20가지의 변환을 수행해야 한다.

현재 TDT v1.0 표준에서 변환될 수 있는 Tag Data의 종류로는 태그 데이터 표준(Tag Data Standards Version 1.1r1.27, TDS v1.1r1.27)에 정의된 스킴 13가지(GID-96,SGTIN-64, SGTIN-96, SSCC-64, SSCC-96, SGLN-64, SGLN-96, GRAI-64, GRAI-96, GIAI-64, GIAI-96, USDOD-64, USDOD-96)가 있다.

이러한 모든 모듈을 직접 수작업으로 코딩하는 것은 시간과 인력이 많이 소모되므로 여기서는 TDS v1.1r1.27에 정의된 각 스킴에 대한 마크업 파일(XML)을 입력으로 받아 해당 변환 모듈 코드를 생성해주는 코드 생성기(code generator)를 설계, 구현하여 변환을 정적으로 수행하는 13가지의 변환 모듈을 생성하고 이를 통합하여 단일 인터페이스를 지원하는 하나의 클래스를 생성하여 태그 데이터 변환을 수행하는 시스템을 설계하고 구현한다.

2. 본론

RFID Tag Data를 변환하는 방법은 크게 동적 변환코드 생성 방식과 정적 코드 생성 방식의 두방식으로 구분할 수 있다. 첫 번째 방법은 TDT XML 파일을 실행시에 읽고 동적으로 분석해 요구된 형식으로 변환을 수행하는 것이다. 두 번째 방법은 변환 코드를 미리 생성하고 입력된 Tag Data를 요구된 형식으로 변환하는 것이다. 첫 번째 방법은 개발 초기에 테스트를 수행하기에 용이하나 Tag Data 변환을 빈번하게 수행하는 경우에 두 번째 방법이 수행효율 측면에서 우수할 것이다. 이 장에서는 정적으로 변환을 수행하기 위해 Tag Data 변환을 수행해주는 정적 코드를 생성하는 코드생성기(Code Generator)를 설계하고 구현한 내용을 기술한다.

2.1 TDT 변환 툴의 시스템 구성

전체적인 시스템의 구성은 코드 생성부와 변환 수행부로 나눌 수 있다. 코드 생성부에는 TDS v1.1r1.27에 정의된 스킴의 마크업 파일을 Code Generator에서 사용가능하게끔 자바클래스로 변환해주는 Scheme Loader와 Scheme Loader에서 변환된 마크업 파일을 분석하여 코드를 생성하는 Code Generator로 구성되어 있다. 변환 수행부에는 자동으로 생성된 변환 모듈을 통합할 수 있는

TDTTranslator와 변환 모듈 전체에서 사용되는 Core API, 13가지의 변환 모듈의 공통된 변환 규칙과 사용 자료들을 일반화한 TDS, Level 클래스와 자동으로 생성된 13가지 변환 클래스로 구성되어 있다.

2.1.1 코드 생성부

코드 생성부는 스킴의 마크업 파일(XML)을 읽어 자바 클래스로 바꾸어 주는 Scheme Loader와 이를 사용해 자바 코드를 생성하는 Code Generator로 구성되어 있다.

Code Generator가 변환 코드를 생성하도록 하기 위해 사용자가 별도의 정보를 입력하지 않고 TDS v1.1r1.27에 정의된 스킴 마크업 파일을 Code Generator에 입력으로 주면 된다. 그러면 Code Generator는 입력으로 들어온 스킴 마크업 파일을 내부의 Scheme Loader를 이용하여 자바클래스로 바꾸어 이를 분석하여 해당 스킴에 맞는 변환을 수행하도록 하는 자바 클래스를 생성한다. 여기서 Scheme Loader는 내부적으로 Scheme, Level, Option, Rule, Field 클래스를 사용하여 스킴 마크업 파일에 정의된 내용을 자바 클래스로 변환하는 역할을 한다.

2.1.2 변환 수행부

변환 수행부는 미리 작성해야 할 부분과 Code Generator에 의해 자동으로 생성될 부분으로 구성된다. 미리 작성해야 할 부분은 TDTTranslator, TDS, Level, CoreAPI 그리고 TDTTranslationException의 하위 예외 클래스들로 구성된다. 자동으로 생성될 부분은 각 스킴마다 생성되는 TDS의 구체화 클래스들이다.

이러한 변환 수행부는 각 스킴들의 변환 과정 중에서 공통적으로 수행되는 부분을 일반화하여 TDS라는 추상클래스로 작성하고, 이를 지원하기 위해 TDS의 구현체에서 내부 level들이 가져야 할 메소드와 변수를 일반화하여 Level이라는 추상클래스로 작성한다. 그리고 태그 데이터 변환 수행 시 자주 자용되는 메소드들을 모아 CoreAPI 클래스로 작성한다.

5. 결론

태그코드 변환 기능은 RFID 기반 시스템의 여러 모듈에서 사용되고 빈번히 수행되므로 태그 변환코드는 수행효율이 우수해야 한다. 또한 새로운 코드 체계와 변환 과정을 손쉽게 추가할 수도 있어야 한다. 본 논문에서는 TDS v1.1r1.27의 마크업 스펙으로부터 변환코드를 자동 생성하는 방식을 제안하였다. 이 방식은 다양한 조합의 변환 코드를 간단히 개발할 수 있게 하고 변환코드의 유지보수가 편리하다. 또한 동적수행 방식의 Accada[4] TDT 모듈과의 실험을 통해 적게는 2배에서 많게는 5배의 우수한 실행 효율을 확인하였다.

참고문헌

[1] EPCglobal. http://www.epcglobalinc.org
[2] EPCglobal, EPCglobal Generation 1 Tag Data Standards Version 1.1 Rev.1.27 Standard Specification, 2005
[3] EPCglobal, EPCglobal Tag Data Translation (TDT) 1.0 Ratified Standard Specification, 1-107, 2006
[4] Accada. http://accada.org/tdt/index.html