원격 측정 장치의 데이터 수집을 위한 데이터 유효성 검사 모듈의 개발

Development of Data Validity Check Module to Collect Data of Remote Measuring Device

부산대학교 컴퓨터공학과
김선영
E-mail : s.y.kim@pusan.ac.kr

Revised at 2010.08.31

ABSTRACT

원격 실험 데이터를 관리하고자 할 때, 실험 데이터를 정확하게 서버로 옮겨 데이터베이스를 구축하면 데이터를 효과적으로 관리할 수 있다. 일반적으로 데이터베이스를 구축할 때는, 데이터의 유효성이 중시된다. 본 연구실에서 개발한 원격 측정 장치를 위한 실험 데이터 관리 시스템은 원격 측정 장치의 데이터를 폴링하고 이를 파싱하여 데이터베이스로 구축하고 있으나, 현재의 실험 데이터 관리 시스템은 데이터가 바른 값과 포맷을 가지는지 확인하지 않기 때문에 적합하지 않은 데이터가 입력되면 시스템의 데이터 무결성이 깨질 수 있다. 따라서 본 보고서에서는 원격 실험 데이터를 정제하여 데이터베이스를 구축하고자, 데이터 유효성 검사 모듈을 생성하였다. 이 모듈을 시스템에 적용함으로써 유효성 검사를 통해 데이터 입력 시스템의 데이터 무결성을 해칠 수 있는 요인들을 제거할 수 있다. 하지만 이미 데이터베이스에 입력된 데이터가 무결한 것인지에 대한 검증 모듈이 반드시 필요할 것으로 판단된다. 추후, 데이터 검증 모듈을 개발하여 서버 단계에서 데이터와 관련하여 발생할 수 있는 오류들을 최소화하고자 한다.

Keywords Remote Data Management, Validity Check Module, Verification Module

1 서 론

원격 측정 장치의 실험 데이터를 관리하기 위해서 실험 데이터를 서버로 옮겨 데이터베이스를 구축할 경우, 데이터를 효과적으로 관리할 수 있다. 데이터베이스를 구축하는 과정에서는 일반적으로 데이터의 유효성이 중시된다. 유효하지 않은 데이터가 입력될경우 데이터의 무결성이 깨지고, 시스템의 안정성을 해칠 수 있기 때문이다. 본 연구실에서 개발한 원격 측정 장비의 실험 데이터 관리 시스템은 데이터를 효과적으로 관리하기 위해 서버에서 데이터베이스를 구축하는 방법을 사용하고 있다. 그러나 현재의 실험 데이터 관리 시스템은 데이터 파일의 포맷이 잘못되었을 경우에만 체크를 하고 있을 뿐, 데이터의 각 속성의 범위나 값의 유효성을 검사하지 않기 때문에 적합하지 않은 데이터가 입력될 경우 시스템의

무결성이 깨질 수 있다[1]. 따라서 본 보고서에서는 원격 실험 데이터를 정제하여 데이터베이스를 구축하고자, 데이터 유효성 검사 모듈을 생성하고 이 모듈을 적용하였을 때의 시스템 성능을 평가하고자 한다.

2 유효성 검사 모듈의 설계

원격 측정 장치의 실험 데이터 값은 주로 수치로 나타난다. 따라서 데이터의 유효성을 검사할 때 각 속성이 나타날 수 있는 범위를 지정하여 이를 벗어나지 않는지 확인해야 한다. 데이터 종류에 따른 속성의 자료형은 표 1과 같다.

데이터 종류	K3000	K3100	K3600
실수형	17	11	21
문자열형	1	9	4
정수형	4	4	1

표 1. 각 데이터 종류에 포함된 속성들의 자료형. 실수형의 속성이 대부분임을 알 수 있다.

표 1 에서 확인할 수 있듯이, 실험 데이터의 속성은 실수 형의 수치가 대부분이고, 문자열 형과 정수형의 값이 낮은 비율로 존재한다. 자료형에 따라 데이터의 특성도 달라지기 때문에, 자료형에 따라 유효성 검사 모듈의 적용을 달리해야 한다. 자세한 내용은 다음과 같다.

a. 실수형 데이터
실수형인 데이터는 대부분 실험으로 얻은 실제 수치이다. 이러한 데이터는 소수점 아래의 숫자가 많이 나타나므로 지수형으로 나타나는 것이 많고, 데이터베이스에도 지수표기법으로 기록한다.

b. 정수형 데이터
정수형 데이터들은 대부분 실험 데이터를 가공하여 얻은 수치이다. 그렇기 때문에 데이터로 반드시 수치만을 얻는 것이 보장되지 않는다. 이들 데이터 값으로는 무한대를 의미하는 'Infinity', 결과가 존재하지 않음을 의미하는 'NaN' 과 같은 문자열 데이터도 입력될 수 있기 때문에 데이터의 무결성을 해칠 가능성이 있다. 따라서 실수형이나 정수형에 입력된 문자열은 표 2와 같이 대치하여 데이터베이스에 입력한다.

입력된 문자열	데이터베이스 저장 값
NaN	-99999.99999
Infinity	99999.99999
의미없는 문자열	-77777.77777

표 2. 실수 데이터에 입력된 문자열을 대치할 값. 데이터베이스에는 입력한 문자열이 아닌 대치한 값을 저장한다.

데이터의 분포를 확인하기 쉽도록 가공한 데이터들은 그 값의 범위가 제한되어 있는데, 대표적

으로 데이터 전체의 평균값이나 효율성 등 percentage 값을 가지는 속성들이 그 예이다. 따라서 각 속성에 따라 최대, 최소 값을 설정해야 할 필요성이 있다.

1. 문자열 데이터
   문자열 데이터들은 날짜와 파일 이름, 장치 ID 등의 고유 값이 많기 때문에, 수치를 값으로 가지는 데이터들과는 달리 데이터의 유효성을 검사해야할 필요는 없다.

3 유효성 검사 모듈의 구현

원격 측정 장치의 실험 데이터를 관리하는 시스템에 데이터의 유효성을 검사하는 모듈을 내장하기 위해서, 시스템을 구축할 때와 동일한 환경에서 모듈을 구현하였다. 개발언어는 Java로, Java SDK 6를 사용하였다.

앞서 설명한 문자열 데이터의 특징 때문에 유효성 검사 모듈은 실수형과 정수형의 데이터에만 적용한다. 실제로 유효성 검사의 주 대상이 되는 데이터는 정수형 데이터이나, Java에서는 실수형 데이터를 간단한 함수를 통해 정수형으로 형변환할 수 있으므로, 입력 데이터로는 실수형만을 고려하였다.
유효성 검사 함수를 생성할 때 고려해야할 사항은 1) 전달인자를 무엇으로 받는지, 2) 반환형이 무엇인지, 3) 예외 처리 해야할 내용은 무엇인가에 대한 내용이다. 1)과 2)는 실수형 데이터만을 전달 인자로 고려하고, 반환형 역시 실수형임을 앞서 언급한 바 있다. 이는 정수형의 경우 Java에서 제공하는 Double(double값).intValue() 함수를 통해 실수형의 데이터를 간단히 정수형으로 변환할 수 있기 때문이다[2]. 3)의 예외 처리 대상은 일반적으로 적용할 수 있는 항목이 아니다. 각 속성의 최대, 최소 값을 환경 설정 파일을 통해 설정한 후, 매번 속성을 확인하여 그 값의 범위를 체크해야 한다. 현재는 각 속성에 대한 자세한 스펙 사항을 알지 못하므로, 상식 수준에서 생각할 수 있는 정도의 유효성만 체크하였다. 가령 Fill Factor 속성은 백분율 값을 가지므로, 0∼100 사이의 값을 가질 것이라고 예측할 수 있다. 위의 내용에 대한 자세한 코드는 그림 1 과 같다.

4 실험

4.1 실험 데이터 및 실험 방법

유효성 검사 모듈에 대한 실험은 해당 모듈 자체만으로 성능을 평가하기보다 실제로 시스템에 적용했을 때 제대로 작동하는지 확인하는 것이 바람직하다고 생각하여, 시스템의 동작 환경과 동일하게 설정하였다. 실험 환경은 Windows 7 OS, MySQL , Java SDK 6 이다.
입력 데이터로는 세 종류의 데이터 K3000, K3100, K3600을 사용하였다. 입력 데이터에 대한 자세한 정보는 표 3 와 같다. Cracked 데이터는 속성 자료형과 일치하지 않는 값이 들어간 것을 의미하는데, 예를 들면 Fill Factor 속성은 백분율 값을 가지기 때문에 실수나 정수형의 데이터가 입력되어야 하나 'NaN'이나 'Infinity', 또는 의미없는 문자열 등으로 입력되었을 경우 이를 cracked data로 여긴다. 다만 'NaN'이나 'Infinity'를 입력받은 경우, 이는 데이터의 무결성을 해칠 수 있지만 상용적으로 사용하고 있는 의미있는 데이터이므로, 각각 -99999.99999, 99999.99999로 대치하여 데이터베이스에 입력하기로

```
public Double isNumber(String input)
{
    ValidCheck vc = new ValidCheck();
    return vc.isNumber(input, Double.NaN, Double.NaN);
}

public Double isNumber(String input, double max, double min)
{
    double value = 0.0;
    try{
        value = Double.valueOf(input);
        if      (Double.isInfinite(value))   { value = 99999.99999; }
        else if (Double.isNaN(value))        { value = -99999.99999; }
    }
    catch(NumberFormatException e){
        value = -77777.77777;
    }

    // min, max 검사
    if (new Double(min).compareTo(Double.NaN) != 0 && new Double(max).compareTo(Double.NaN) != 0)
    {
        if (value > max )
        {
            System.out.println("NumberRangeExceptionError : The value is higher than the MAXIMUM Value");
        }
        else if (value < min)
        {
            System.out.println("NumberRangeExceptionError : The value is lower than the MINUMUM Value");
        }
    }

    return value;
}
```

그림 1. 유효성 검사 모듈의 내장 함수

한 바 있다. 이를 제외한 의미없는 문자열은 -77777.77777로 대치하여 입력한다. Cracked 데이터의 대표적인 예는 그림 2와 같다.

데이터 종류	데이터 유형	데이터 개수[Cracked/정상]
K3000	csv	3/4
K3100	xls	10/49
K3600	csv	2/15

표 3. 실험에 사용한 데이터의 종류와 개수. 정상 데이터와 오류가 있는 데이터를 혼합하여 사용하였다.

실험은 세 가지 경우로 나누어 1) 정상 데이터를 파싱하였을 때, 2) 정상 데이터와 cracked 데이터를 혼합하여 파싱하였을 때, 3) cracked 데이터를 파싱하였을 때 각각 시스템의 종료 없이 오류를 보고하면서 파싱을 수행하는지에 대해 실험하였다.

Voc :	0	V
Voc_Init :	-0.011	V
Isc :	0	mA
Isc_Init :	0	mA
Jsc :	-3.8E-05	mA/cm^2
In_Power :	98.2	%
Fill Factor :	NaN	%
Pmax :	0	mW
Vmax :	0	V
Imax :	0	mA
Efficiency :	0	%

그림 2. Cracked data의 예. Fill Factor 속성은 백분율 값을 갖기 때문에 실수나 정수형의 데이터가 입력되어야 하나 'NaN'으로 입력되어 있다. 'NaN' 값도 Fill Factor의 속성값으로는 유효하나, 데이터베이스로 구축하면 데이터의 무결성을 깨므로 데이터베이스로 입력 불가능하다.

4.2 실험 결과

예상한 실험 결과는 데이터 파싱 시 적합하지 않은 형식의 값을 가지는 데이터를 자동으로 적절한 값으로 변환하여 시스템의 종료 없이 데이터베이스에 저장하는 것이었다. 실험 결과, 오류를 포함한 모든 데이터들의 파싱에 대해 성공하였다. 자세한 내용은 표 4 과 같다.

데이터 종류	데이터 개수	오류 보고 개수	파싱 여부
정상 데이터	19	0	성공
정상 데이터 + cracked data	19 + 15	15	성공
cracked data	15	15	성공

표 4. 정상 데이터와 cracked 데이터

정상 데이터는 데이터베이스에 그대로 구축하면 되지만 cracked 데이터의 경우에는 데이터베이스 입력 자체가 불가능하다. 따라서 앞서 cracked 데이터는 각각에 해당하는 다른 수치로 대치하여 입력하기로 했었다. Cracked 데이터를 포함하고 있는 원본 데이터와 이 데이터가 데이터베이스에 저장된 모습은 그림 3와 같다.

5 결론 및 추후연구

원격 측정 장치의 실험 데이터를 관리하는 시스템에서는 데이터를 폴링하고 파싱하여 데이터베이스를 구축하는 작업이 우선적으로 필요하다. 이 때, 데이터의 무결성을 지키고 시스템의 안정성을 위하여 데이터의 유효성을 검사하는 과정은 필수적이다. 그러나 지금까지 개발한 원격 실험 데이터 관리 시스템에는 이 과정이 누락되어 있기 때문에, 적합한 포맷의 데이터가 아니거나 데이터의 값이 유효한

System Time	Life Time	Volt	Current(m	Temp	Humi(%)
2010.04.19_18:47:13	abcd	NaN	Infinity	25	10.6
2010.04.19_18:47:13	0.005	0.115	0	25	10.6
2010.04.19_18:47:13	0.005	0.117	0	25	10.6
2010.04.19_18:47:13	0.005	0.119	0	25	10.6
2010.04.19_18:47:13	0.005	0.12	0	25	10.6

(a) 원본 cracke data

Idx	K3600MI_Idx	K3600MR_Idx	SysTime	LifeTime	Volt	Curr	Temp	Humi
1	12	1	2010.04.19_18:47:13	-77777.77777	-99999.99999	99999.99999	25	10.6

(b) cracked data가 저장된 모습

```
Error 1 : NumberFormatException - character
Warning 2 : NumberFormatException - NaN value
Warning 1 : NumberFormatException - Infinity value
Warning 2 : NumberFormatException - NaN value
Warning 1 : NumberFormatException - Infinity value
```

(c) cracked data에 대한 에러 보고

그림 3. Cracked data를 포함하고 있는 원본 파일 (a)와 이것이 저장된 데이터베이스 (b)의 모습. 실수 데이터를 입력받아야 할 Life Time, Volt, Current 속성에 각각 abcd, NaN, Infinity 문자열 값이 입력되었으므로 유효성 검사모듈에서 걸러야 한다. (b)의 데이터베이스에 저장된 모습을 보면, 'NaN'은 -99999.99999로, 'Infinity'는 99999.99999로, 그 외의 의미없는 문자열은 -77777.77777로 대치된 것을 확인할 수 있다. (c)는 cracked data가 저장될 때 시스템이 보고한 에러 리포트이다. abcd는 의미없는 문자열이므로 1번 Error, NaN과 Infinity는 각각 2번 Warning, 1번 Warning으로 보고했음을 알 수 있다.

범위를 벗어날 경우 데이터 무결성을 해칠 수 있다. 따라서 본 보고서에서는 원격 실험 데이터의 유효범위와 값을 설정하여, 검증한 데이터만으로 데이터베이스를 구축할 수 있도록 데이터 유효성 검사 모듈을 생성하였다. 사용자가 각 속성의 범위를 설정하면, 시스템이 데이터를 파싱하면서 오류가 있거나 유효하지 않은 데이터를 판단하여 안정적으로 오류를 검출하고 기록할 수 있었으며, 이로 인해 시스템의 안정성도 도모할 수 있었다. 하지만 데이터베이스에 입력된 데이터가 무결한 것인지에 대한 검증 모듈이 반드시 필요할 것으로 판단된다. 추후, 데이터 검증 모듈을 개발하여 서버 단계에서 데이터와 관련하여 발생할 수 있는 오류들을 최소화하고자 한다. 또한 현재는 발생할 수 있는 몇 가지 에러상황에 대해 임의로 Warning과 Error로 나누어 에러 보고를 했으나, 차후 보고내용을 Error로 통일하고, 이를 시스템 출력이 아닌 log로 남겨 Error의 원인과 발생위치를 한눈에 파악하기 쉽게 하며, Error 번호만으로 Warning 여부를 판별할 수 있도록 통일성을 부여할 예정이다.

참고 문헌

1. Kim SeonYeong, "Tcp/ip 소켓을 이용한 원격 측정 장치와 실험 데이터 통합 관리 시스템 개발," *Technical Report*, 2010.
2. Java, "Java api," http://download.oracle.com/javase/7/docs/api/index.html?overview-summary.html.