

# 레거시 시스템 확장을 위한 XML 프레임워크

- 분산 환경의 에러를 고려한 자동화 시스템 구현 -

임성현\*, 최진영\*\*

\*고려대학교 컴퓨터과학기술대학원 소프트웨어공학과

\*\*고려대학교 컴퓨터학과

e-mail : [choi@formal.korea.ac.kr](mailto:choi@formal.korea.ac.kr), [allblue1013@yahoo.co.kr](mailto:allblue1013@yahoo.co.kr)

## XML Framework for Legacy Extension

-Implementation of Automation System considering Distributed Environment Error-

Sung-Hyun Lim\*, Jin-Young Choi\*\*

\*Dept of Software Engineering, Graduate Schools of Computer Science & Technology, Korea University

\*\*Dept of Computer Engineering, Korea University

### 요 약

기업 시스템의 통합은 기업의 경쟁력을 향상시킬 수 있고, 기업 시스템에 웹 시스템을 접목시킴으로 기업의 활동 영역을 넓힐 수 있다. 이에 본 논문에서는 FLX (Framework for Legacy Extension) 을 제안했다. 주어진 절차에 따라 레거시 시스템은 전송을 요청하고, FLX 시스템은 이를 주기적으로 검색해서 웹 환경의 시스템으로 전송하고 그 결과를 확인하고 정상 전송 여부를 남기게 된다. 본 시스템의 특징은 통합에 표준적인 XML 통합 방법을 제안 했고, 동기 통신과 비동기 통신의 혼합형으로 레거시 측면은 비동기 통신, 내부적으로는 동기 통신을 이용했다. 이는 레거시 시스템의 지연을 방지해 주고, 전체적인 신뢰도를 높이는 특징이 있다. 또한 네트워크 에러인 경우 자동으로 재전송 하고, 전송하고자 하는 데이터의 에러인 경우 로그 정보를 남겨, 관리할 수 있도록 했다.

### 1. 서론

현재까지 기업의 비즈니스 로직을 포함하고 있는 기간 시스템은 인터넷의 등장과 함께, 이를 통한 기업의 효율성 증대를 위한 요구를 받게 된다. 또한 기존까지는 필요에 따라 그에 적절한 시스템을 제공하는 방식에서 데이터의 중복성 제거와 기업 전반을 포괄하는 뷰를 제공하기 위하여, 통합된 시스템의 관심 또한 늘어나고 있다.

하지만, 지금의 각각의 시스템을 새로운 하나의 통합된 시스템으로 옮기는 작업은 새로운 비용이 소요된다는 측면 외에도 지속적인 변경이 이루어진 내부 전체를 최신 현황으로 파악해야 한다는 점에서 매우 부담이 크다.

이에 기존의 시스템간을 통합하는 노력이 있게 되고, 다양한 방법과 다양한 솔루션들이 등장하게 되며, 현재 운용 중인 시스템을 이용해야 하므로 단일의 에러는 곧 기업 기간 시스템의 불안정으로 연결되는 바, 매우 신중하게 다루어 져야 한다. 곧, 표준화된 프레임워크가 필요한 것이다.

본 논문에서는 시스템간의 통합 방법을 소개하고, 각 방법들의 문제점과 동시에 요구 사항에 부합하는 방법을 제시하여 설계와 구현과 테스트를 통해 안정성을 검증 해 보겠다. 본 논문의 구성으로는 2 장에서는

레거시 시스템과 시스템 통합에 관련된 연구 및 기술을 소개하고, 이의 문제 제기과 해결의 기본 방법을 제시하고, 3 장에서는 제안하는 시스템 개요와 이에 대한 상세 설계를 설명하며, 4 장에서는 제안한 시스템을 테스트 하고 그 결과에 따른 검증을 보여주며, 5 장에서는 결론 및 향후 연구 과제를 소개한다.

### 2. 관련연구

레거시 시스템은 이제는 사용되지 않거나, 시대에 뒤떨어진 시스템 아키텍처를 사용하고, 외부 환경과 상호 운영하는 인터페이스를 가지고 있지 않을 가능성이 높고, 전반적인 교체 및 수정하기에 용이하지 않는 반면, 내부 요청에 의한 수정 보완이 계속 지속되며, 또한 현재 사용 중이고, 유용하지만, 가까운 미래에 신규 시스템으로 대체될 예정이 라는 특징을 가지고 있다. [2] [5]

최근의 신규 프로젝트는 외부와 단절된 시스템(stand-alone system)을 구축하는 프로젝트는 찾아볼 수 없고, 외부와 지속적인 연동과 동시에 내부의 기존 시스템과도 연동되는 통합된 시스템(integrated system)을 구축해야 하는 필요성을 가지게 된다. 이는 과거의 고립되고 매우 제한된 상호운영성만을 가지던 환경에서 각 영역의 정보를 비즈니스 프로세스로 통합하여 기업

이율을 극대화 할 수 있게 된 것이다. 이에 따라 기업의 내부 시스템의 통합의 필요성이 증가되었다. [1][3] 이와 같은 통합의 필요성과 요구사항에 따른 기존의 통합 방법을 보면 다음과 같다.

- 1) 메시지 전달 방식으로서의 분류
  - 동기 / 비동기[7][8][10]
- 2) 아키텍처로서의 분류
  - 데이터 베이스 링크 / 미들웨어 / EAI (Enterprise Application Integration)[5][10][11]

하지만, 기존의 통합 방법에는 다음과 같은 문제점을 가지고 있다.

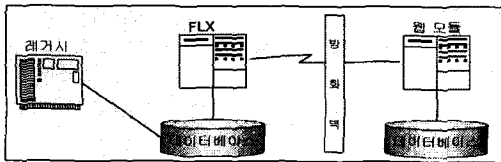
- 1) 신규 시스템 도입에 따른 부담
- 2) 벤더 의존도 증가
- 3) 수정 추가때마다 메시지 추가 정의 필요
- 4) 데이터베이스간의 연동시 방화벽을 사이에 둔 시스템에 부적합
- 5) 호출한 후 그 결과 값이 올 때까지 다른 작업을 할 수 없는 동기화 통신과 호출한 측에서 그 결과를 바로 확인할 수 없으며, 100% 신뢰도를 보장하지 않음으로 인해 시스템 불안정 시 자료 손실의 가능성을 가지고 있는 비동기 통신 중 택일해야 함.
- 6) 안정적인 대용량 전송을 지원하는 방법이 없음.

따라서 본 논문에서 거론한 문제점에 대한 해결 방안으로 다음과 같은 기본 목표를 제안한다

- 1) XML over HTTP 사용[4] [6] [9]
- 2) 정상/비정상 전송의 로그 기록
- 3) 동기/비동기의 통합
- 4) 전송 요청부터 결과 처리까지 프레임워크 제공
- 5) 최대 전송량을 설정해 대용량 전송을 위한 방법을 제시

### 3. 설계

제안하는 시스템의 명칭으로는 FLX (Framework for Legacy Extension)으로 명명했다. FLX 시스템의 구성도는 <그림 1>과 같다.



<그림 1> 시스템 구성도

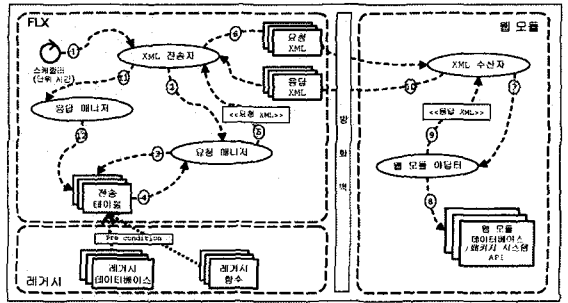
FLX 시스템의 내부 흐름은 다음과 같다.

XML 전송은 단위 시간을 간격으로 전송을 구동시키는 스케줄러에 의해 시작된다①. 구동된 XML 전송자는 요청 매니저를 생성한다②. 요청 매니저는 전송테이블을 검색하고③ 전송할 데이터가 있는 경우④ 요청 XML 문서를 생성해서 XML 전송자에게 전달한다⑤. XML 전송자는 웹 모듈의 XML 수신자에게 요청을 HTTP POST 로 전송한다⑥.

XML 수신자는 전달 받은 요청 XML 을 웹 모듈 어댑터를 통해⑦ 실제의 웹 모듈의 데이터 베이스나 API

를 통해 처리하게 된다⑧. 처리 결과에 따라 웹 모듈 어댑터는 응답 XML 을 생성하고⑨ XML 수신자는 다시 XML 전송자에게 회신하게 된다⑩. 처리 결과 정보를 가지고 있는 응답 XML 을 XML 전송자가 응답 매니저에게 전달하고⑪, 응답 매니저는 결과를 전송테이블에 업데이트 하면서⑫ 한 프로세스가 종료된다.

<그림 2>는 본 설명을 표현한 조감도 이다.



<그림 2> 시스템 조감도

FLX 시스템의 상세 설명을 살펴보면 다음과 같다.

1) XML 생성 및 데이터 전송  
시스템 사이의 데이터 송수신을 위한 기본적인 형식으로는 XML 을 이용하고, 전송되는 XML 의 최대전송량을 제한함으로써, 순간적으로 대량의 데이터 전송이 요청될 때 분할 전송을 통해 시스템의 과부하를 막았다.

2) 전송 테이블 - 구조  
전송 테이블은 크게 실제 전송할 데이터를 가지는 데이터 영역과 전송 및 에러 처리를 담당하는 컨트롤 영역으로 나뉘었다. 전송 테이블은 한 item 의 신규/수정/삭제를 한 데이터로 관리하고, 전송중에는 데이터 수정을 막는 제한을 갖도록 해서 데이터의 생성/수정/삭제를 제어할 수 있도록 했다.

컨트롤 영역에 포함되는 주요 내용으로는 PROCESS\_CD, ACTION 등이 있으며, 컬럼의 설명은 다음과 같다.

- PROCESS\_CD(전송 진행 코드)  
진행 코드는 데이터 전송의 진행 내역을 담게 되며, 전송되는 데이터는 [N]상태와 [F]상태이고, 전송이 진행되면 [P]상태로 바뀌어, 전송 중 임을 명시 하며, 동시에 데이터의 변경을 제한한다. 전송 결과에 따라 정상 전송 완료([C]), 전송 실패([F],[W])로 상태가 변경된다. 다음 [표 1]은 PROCESS\_CD 에 따른 설명을 다시 표로 요약한 것이다.

[표 1] PROCESS\_CD 설명

코드	설명
N	신규 데이터
P	전송 중
C	정상 전송 완료
F	네트워크 문제로 재전송되어야 함
W	데이터 내부의 문제로 재전송되지 않음

- ACTION

ACTION 은 데이터의 처리 요청을 의미하며, 각기 I (생성)/U(수정)/D(삭제)를 의미한다.

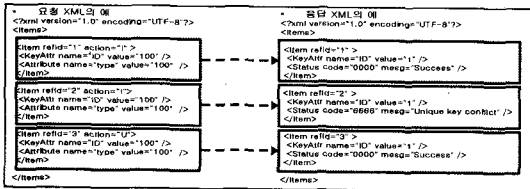
3) 데이터 에러처리

네트워크의 불안정이나, 웹 모듈의 서버 문제인 경우, PROCESS\_CD 를 F 로 변경 하고 재 전송하고, 데이터 자체의 문제인 경우 PROCESS\_CD 를 W 로 변경하고 관리자가 확인할 수 있도록 했으며, W 의 에러 내용은 다음과 같다.

- 중복된 신규 요청(unique key violation)
- 수정/삭제할 원본이 없음(update/delete fail)
- 잘못된 데이터(invalid data)
- 기타 에러.

4) 요청/응답 XML

요청 XML 에 따른 결과는 응답 XML 에 담겨 전달된다. <그림 3>은 XML 의 예제로써, 요청에 대한 응답 XML 의 연관 관계를 보여주고 있다.



<그림 3> 요청/ 응답 XML 의 예

4. 시험 및 평가

제안한 시스템을 구현하기 위해 다음과 같은 환경이 적용되었다.

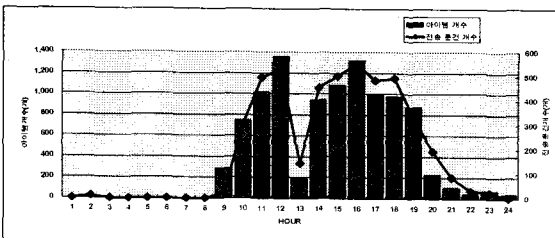
[표 3] 구현 환경

구분	내용
운영 체제	Sun Solaris 8
서버	Sun Enterprise Server 급
개발 언어	Sun JAVA 1.3.1_04
레저시 시스템	In-house made (4GL)
WAS	iPlanet Web Server 5.0 iPlanet Application Server 6.0
웹 솔루션	사사의 i 솔루션

또한 본 시스템을 평가하기 위해 필드 테스트를 실시했다.

전송을 위한 환경으로, 전송 간격은 15 초, 1 회 최대 전송 제한량은 200item, 테스트는 20 일간 29,000 item 에 대해 이루어 졌다. 이 중 네트워크 에러로 인한 에러는 172 건, 데이터 자체의 오류로 인한 에러는 45 건 발생하였다.

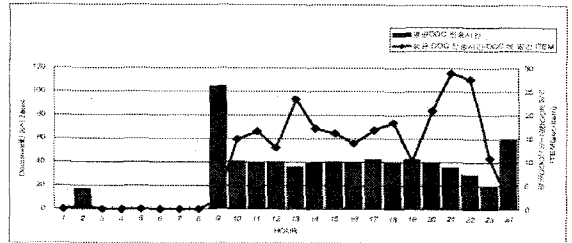
다음 <그림 4>는 전송하는 데이터의 속성이다.



<그림 4> 데이터의 속성

본 데이터의 속성을 보면, 전송 요청은 일과중에 크게 일어나며, 전송 문건 수 또한 동일하게 증감함을 볼 수 있다.

다음 <그림 5>는 전송에 소요된 시간이다.



<그림 5> 전송 소요시간

본 필드 테스트 결과 시스템 자체가 안정적임을 볼 수 있었다. 즉, 전송되는 양이 전송을 요청하는 양을 전부 처리하였다는 의미이다. 즉, OUTPUT > INPUT 인 경우 안정적이라는 의미이다.

또한 OUTPUT > INPUT 의 가정을 FLX 시스템에 적용해

FLX 가 단위시간 내 처리하는 소요 시간 < 단위시간 ... (1)

라는 식을 세울 수 있고,

소요시간 = 재전송계수 \* 안전계수 \* (document 생성 전송시간 + 건당 소요 시간 \* item) ... (2)

와 같은 식을 유도할 수 있다. 이에 따른 주요 설명은 다음과 같다.

- 재전송 계수 ( $r > 1$ )  
네트워크의 불안정으로 재전송되는 비율
- 안전 계수 ( $k > 1$ )  
예측한 양보다 더 많은 양의 요청이 들어올 때의 수용을 고려한 비율
- document 생성 및 전송시간 (DT)  
전송되기 전에 전송테이블을 검색하고, 네트워크상에서 전송을 위해서 소요되는 시간 등 전송되는 item 의 양과 상관 없이 소요되는 고정 시간
- 건당 소요시간 (SI)  
전송되는 item 양이 증가됨에 따라 변동이 이루어지는 시간으로, 본 테스트 환경에서는 선형으로 증가함을 볼 수 있었다. 따라서, 본 건당 소요시간은 전송 요청 item 수에 대한 전송 소요시간을 일차 선형 회귀 분석을 통해 실험적으로 파악할 수 있다.
- 전송 요청 item 수 (item)  
단위 시간 내에 요청된 item 수를 document 의 최대 전송 허용량이 고려해서 한 document 에 담기게 되는 item 수

이에 따라 단위 시간 내에 최대 전송량을 전송하는데 소요된 시간을 파악하면

최대 전송량 전송 소요시간 = 재전송 계수 \* 안전계수 \* [건당 소요 시간 \* peak 시간대의 item 수 + (document 생성 및 전송시간 + 시간 간격) \* document 전송 회수] < peak 시간의 단위 시간 ... (3)

다시 도입된 식(3)에 사용된 변수와 상수의 설명을

보면,

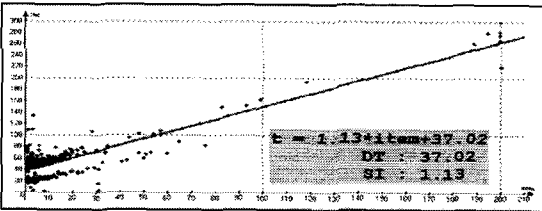
- 시간 간격 (IT)  
스케줄러가 호출되는 시간 간격
- peak 시간대의 item 수 (peaki), peak 시간의 단위 시간(UT)  
FLX 시스템의 안정성을 검증하기 위하여 피크 시간의 전송 요청과 단위 시간을 수치화 시킨 값
- 요청 분산도( $0 < d \leq 1$ )  
단위 시간 안에서 전송이 얼마나 단위 시간 내에서 고르게 분포되어있는지를 판단하는 수치
- document 전송 회수  
해당 내용은 다음과 같이 풀어볼 수 있다.

$$\text{올림}(\text{단위 시간 내 최대 전송 회수} * \text{분산도}) = \text{올림} \{ (\text{peak 시간의 단위 시간} - \text{건당 소요시간} * \text{peak 시간대의 item 수}) / (\text{document 생성 및 전송시간} + \text{시간 간격}) * \text{분산도} \} \dots (4)$$

식(3)에 변수 명을 적용한 식이 다음과 같다.

$$r * k * [SI * \text{peaki} + (IT + DT) * \text{올림} \{ (UT - SI * \text{peaki}) / (IT + DT) * d \}] < UT \dots (5)$$

이에 필드 테스트 자료를 가지고 1 시간당 1,000 건의 요청(요청 분산도  $d=0.6$ )에 대한 안정성을 검증해보겠다. 선형 회귀를 위한 데이터: 4653 document



<그림 6> 테스트 자료에 대한 선형 회귀

- 선형 회귀 결과  
 $DT = 37.02 \text{ 초} \mid SI = 1.13 \text{ 초}$
- 설정한 환경 변수  
 $r = 1.02 \mid k = 1.2 \mid IT = 15 \text{ 초}$
- 검증할 데이터  
 $\text{peaki} = 1,000 \text{ item} \mid UT = 3,600 \text{ 초} \mid \text{요청 분산도 } d = 0.6$
- 계산 결과

$$1.02 * 1.2 * [1.13 * 1000 + (15 + 37.02) * \text{올림} \{ (3600 - 1.13 * 1000) / (15 + 37.02) * 0.6 \}] = 3283 < 3600 \dots (6)$$

식(6)에 따라 본 예제는 한 시간에 1,000 건의 요청을 안전하게 전송 할 수 있음이 검증 되었다.

본 방법을 통해 안정성을 검증함으로써 현재 시스템의 안정성을 검증할 수 있고, 또한 환경이 변경되어 요구 사항이 변경되었을 때에도 계수 변경과 안정성의 판단 기준을 제시할 수 있었다.

## 5. 결론

기업 시스템의 통합은 기업의 경쟁력을 향상시킬 수 있고, 기업 시스템에 웹 시스템을 접목시킴으로 기업의 활동 영역을 넓힐 수 있다. 이를 위한 시스템 통합의 방법은 여러 가지가 있지만, 통합 자체를 위해 또 다른 시스템을 도입한다는 것 자체부터 부담을 가지게 된다.

그리고, 또 다른 곤란스러운 문제 하나는 통합에 사용되는 메시지 전달 방식으로 동기 방식과 비동기 방식이 있고, 양자 중 택일을 해야 하는 이들은 서로 상반되는 장단점을 가지고 있다는 것이다.

이에 본 논문에서는 다음과 같은 특성을 가지는 FLX (Framework for Legacy Extension)를 제안했다.

- 1) 레거시 시스템과 웹 환경의 시스템간의 표준적인 XML 통합 방법을 제안
- 2) 표준적인 framework 를 통해 비즈니스 로직의 변화가 발생하더라도 변경이 용이한 방법 제시
- 3) 동기 통신과 비동기 통신의 혼합형
  - 레거시 측면 : 비동기 통신( 응답이 올 때까지 기다릴 필요 없음 )
  - FLX 내부적 : 동기 통신( 신뢰도 향상 )
- 4) 로그를 통한 예러 관리

그리고, FLX 시스템에 대한 평가 작업을 실시하였으며, 구현한 시스템의 안정성을 검증하는 식을 유도하고 필드 테스트의 자료를 통해 검증했다.

그러나, 동기 시스템의 원천적인 문제인 데이터베이스 Lock 과 전송하고 결과를 기다리는 지연 시간이 있는 등의 문제점은 가지고 있는 등의 단점이 있으므로 각 환경에 맞는 구현이 필요하겠다. 또한 통합을 시스템적으로만 접근하는 것이 아니라, 비즈니스 측면도 고려해야 하며, 이를 위해 비즈니스 워크 플로우를 시스템 통합 내부에 담는 노력이 많이 이루어지고 있으며, 본 논문의 추후 과제로서도 비즈니스 워크 플로우를 담을 수 있는 시스템이 필요하겠다.

## 참고문헌

- [1] Marinos Themistocleous and Zahir Irani, Evaluating and Adopting Application Integration: The Case of a Multinational Petroleum Company, Proceedings of the 35<sup>th</sup> Annual Hawaii International Conference on System Science, 2002
- [2] XML Solutions, XML White Paper Legacy System Integration Using XML
- [3] Eric Roch, Application Integration: Business and Technology Trends, eAI Journal, August 2002
- [4] JP Morgenthal, XML for Data Integration, eAI Journal, October 2001
- [5] Matjaz B. Juric, et al., Professional J2EE EAI, Wrox, 2001
- [6] Kurt A. Gabrick and David B. Weiss, J2EE and XML Development, Manning, 2002
- [7] Simon St. Laurent, Edd Dumbill, Joe Johnston, Programming Web Services with XML-RPC, O'Reilly & Associates, 2001
- [8] Paul Giotto, et al., Professional JMS Programming, Wrox, 2000
- [9] W3C, eXtensible Markup Language (XML), <http://www.w3.org/XML>
- [10] 전유부, e-Business 를 위한 메시지 지향 미들웨어, 고려대학교 산업정보대학원 전자컴퓨터공학과, 2002 년
- [11] 홍정기, EAI 구현전략과 사례: 통합과 협업을 위한 필수 프로젝트, 시사컴퓨터, 2002