

레거시 시스템 데이터의 실시간 모니터링 시스템 개발 -제조업 생산 현장을 중심으로-

이재호*·남호기*·유우식*
*인천대학교 공과대학 산업경영공학과

Real-time monitoring system of the legacy systems data -Focused on Manufacturing Shop Floor-

Jae-Ho Lee* · Ho-Ki Nam* · Woo-Sik Yoo*

*Department of industrial and Management Engineering, Incheon National University

Abstract

As the development environment is changing with the development of information communication technology, the systems that were used by each service became used with integration. In the process of integrating from existing legacy systems to new system, it should be smoothly integrated or shared, however, it cannot help holding existing technology or component due to significant cost burden for conversion.

In this paper, it was not only classified by types with analyzing the various elements that make up legacy system but an approach and monitoring system were developed to each type. After System application results, data's information generated in each process is provided to other system in real time, so that it has not only secured the work efficiency and reliability but also it is made possible by integrating data in various formats for efficient data management, rapid search and tracking to history. With real-time monitoring system developed in this study, It can be very useful in a variety of industries which require real-time monitoring of distributed legacy system data.

Keywords : Legacy System, Big Data, Real Time Monitoring, Equipment Interface, MES

1. 서 론

오늘날 정보 시스템은 기능과 용도에 따라 진화를 거듭하고 있다. 정보시스템은 대부분 각 업무분야에 맞게 개별적으로 구축되어 사용되고, ICT(정보통신기술: Information and Communications Technology)의 발전으로 개발환경이 크게 변화되고 점차 진화하여 시스템을 통합하여 사용하게 되었다. 이러한 통합과정에서

기존에 운영되던 시스템에서 발생하는 결과물들은 새로운 시스템에 원활히 통합되거나 공유 되어야 하지만 프로그램 소스의 유실, 전문 개발자의 부재, 과도한 통합 비용의 발생 등 다양한 원인으로 통합에 어려움이 있다. 또한 현장의 물리적 장비, 프로세스의 변경이 빈번히 발생하지만 그 결과물들이 시스템에 원활히 통합되지 못하거나 결과를 필요로 하는 곳에 즉시 제공되지 못해 시스템의 효율성이 떨어지는 원인이 되기도 한다.

본 논문은 인천대학교 2013년도 자체연구비로 수행 됨

†Corresponding Author : Ho-ki Nam, Department of industrial and Management Engineering, Incheon National University E-mail: ipics@inu.ac.kr

Received December 28, 2015; Revision Received March 16, 2016; Accepted March 21, 2016.

대부분의 기업들은 중요한 업무를 처리하는데 있어 기존 시스템이란 IT에서 일반적으로 레거시 시스템(Legacy System)이라 정의하고 있다. 레거시 시스템은 낡은 기술이나 방법론, 컴퓨터 시스템, 소프트웨어 등이 과거로부터 내려온 것들을 의미한다. 이는 현재까지 남아 쓰이는 기술을 부른 말일 수도 있지만, 더 이상 쓰이지 않더라도 현재의 기술에 영향을 주는 경우도 포함한다. 레거시 시스템과 그 결과물들을 가지고 있다. 새로운 시스템으로 개발하여도 기존에 운영되던 기술이나 부품으로 계속 유지해야 하는 경우도 있고, 안정성 등의 조건을 충족시키면서 새로운 시스템으로 개발하고 전환하기 위하여 부담해야 하는 막대한 비용 또한 무시할 수 없기 때문에 레거시 시스템을 유지할 수밖에 없는 이유가 되기도 한다. 이런 이유들로 인해 레거시 시스템의 활용은 대단히 중요하다.

레거시 시스템을 활용하기 위해서는 기존 하드웨어 또는 소프트웨어에 인터페이스하기 위한 애플리케이션이나 하위 호환성을 구현해줘야 한다. 구현하기 위한 상용화된 제품들이 있지만, 각각의 용도에 맞춤형 되어 있어 사용자 요구에 맞게 변경하거나 적용하는 데에 많은 시간과 비용이 든다는 어려움이 있다.

레거시 시스템 데이터의 변화를 감시하기 위해서는 대상의 식별(Target Manager), 결과물(이하 데이터)에 연결(Connection), 변화의 추적(Tracking), 수집(Acquisition)을 포함하는 실시간 모니터링 시스템이 필요하다. 대상의 식별은 물리적 대상과 그 곳에서 발생하는 데이터들을 분석하여 유형별로 분류하고, 데이터 연결은 데이터 유형에 맞는 연결방식을 이용하여 감시서버와 연결한다. 추적은 감시 컴퓨터와 연결된 대상에서 발생하는 데이터의 변화여부를 조사하여 수집으로 정보를 전달한다. 수집은 추적에서 전달된 정보를 기초로 결과물에 접근하여 복사하거나 동기화하고 파싱(Parsing)과 매핑(Mapping)을 통해 Legacy Integration Store에 저장하는 역할을 한다. 이러한 과정에서 발생하는 물리적 대상과 데이터들의 연결 상태 또는 오류 기록 등에 대한 정보를 화면을 통해 모니터링 하게 된다.

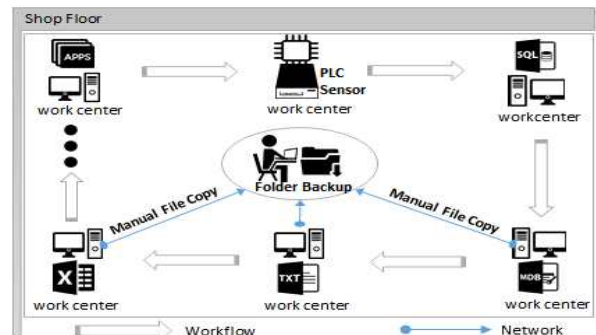
관련 연구들을 살펴보면 다음과 같다. 대용량 관계형 데이터베이스를 시스템간 일괄처리 방식으로 데이터 셋을 분할하여 다중 스테드를 이용하여 수집한 연구가 있으며[4], 설비와 계측장비에서 발생한 대량의 데이터베이스를 실시간으로 수집하여 오라클 데이터베이스에 저장하고, 주기적으로 일괄처리 방식을 이용하여 하둡 파일 시스템으로 저장한 연구가 있다[5]. 또한 철강제조 분야의 공정중 발생한 데이터가 특정 메모리 번지에 마이크로 데이터 형식으로 저장되어 있으며, 일정

주기로 메모리에 저장된 내용을 수집하여 데이터베이스화 한 연구가 있다[6]. [7,8,9]에서는 설비의 상태정보를 PLC인터페이스와 HMI를 이용하여 데이터를 수집 및 데이터베이스화 하여 작업지시와 연계하여 웹서비스 하는 연구를 진행하였다.

본 논문에서는 레거시 시스템의 사용목적, 대상(장치, 응용 프로그램, 저장위치, 데이터의 유형)들을 한정적으로 특정하지 않고, 다양한 대상들의 상태 변화를 실시간으로 조사하고 감시할 수 있는 모니터링 시스템을 소개한다. 감시의 대상을 식별하기 위하여 로컬 또는 원격에 존재하는 레거시 시스템을 물리적 장치와 그곳에서 발생하는 다양한 포맷의 데이터들을 식별 가능한 유형별로 분류하고 식별대상에 대한 속성 정보를 기록 관리한다. 상태의 변화를 조사하기 위하여 분류된 각 대상들에 대한 연결과 추적의 방법을 소개하고, 설계된 감시 시스템의 구조와 각 구성 모듈, 모니터링과 현장의 시스템 적용 결과를 설명하였다.

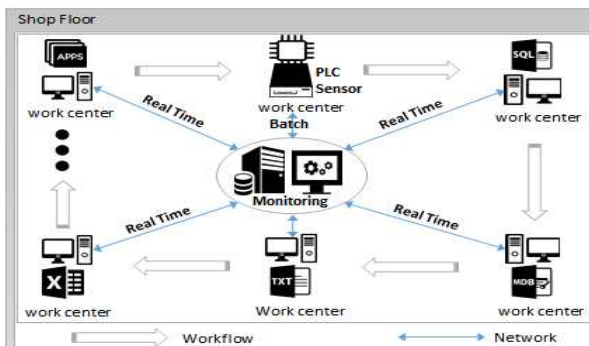
2. 연구의 목적

생산 현장의 각 공정에서 발생하는 생산실적과 품질 검사 데이터 등은 실시간으로 수집되어 연관된 상위 시스템으로 정보를 제공하여 의사결정을 지원해야 하지만, 서론에 기술된 바와 같이 다양한 경우로 인하여 실시간 통합이 되지 못하고 각각의 장치에 장기간 저장되거나 사장된다. [Figure 1]은 작업장에서 발생한 데이터가 사람에 의해 네트워크를 통하여 저장된 데이터를 취합하여 지정된 컴퓨터의 폴더에 저장하는 모습을 보여주고 있으며, Apps(Applications) 또는 PLC/Sensor 와 같은 독립적으로 운영되는 장치들과 데이터 베이스에 저장된 데이터를 수작업에 의존하는 것에는 한계가 있다. 최근에는 랫트에 대한 검사정보 및 생산 이력 정보의 요구가 늘고 있지만 데이터가 통합되어 있지 않아 추적에 어려움으로 즉시 대응하지 못하고 있다.



[Figure 1] As Is

[Figure 2]는 각 작업장에서 발생하는 데이터의 실시간 변화를 감시하기 위한 독립적인 To Be 시스템의 모습이다. 모니터링 시스템은 각 작업장의 장치에 네트워크를 통한 직접 접근 또는 응용 프로그램과 통신하여 장치 및 데이터의 변화를 감시한다. 장치와 데이터의 상태를 사용자 화면 [Figure 8]에 제공하고 데이터를 수집하여 통합 데이터 베이스에 저장한다. 수집 저장된 데이터는 룯트에 대한 생산 이력 정보 요청시 즉시 조회 [Figure 9] 될 수 있다.



[Figure 2] To Be

3. 시스템 분석

3.1 레거시 시스템의 구성

레거시 시스템은 일반적으로 물리적인 장치, 결과를 생성하는 응용 프로그램, 결과물인 데이터로 구성되어 있다. 레거시 시스템의 변화를 감시하기 위하여 대상을 식별하고 분류할 필요가 있으며, 이는 분류된 대상의 유형에 따라 연결과 감시의 방법이 달라지기 때문이다.

① 장치(Physical Device)

장치는 운영체제를 갖추고 통신이 가능한 로컬 또는 원격에 존재하는 컴퓨터, 자동화 장비, 검사장비, PLC(Programmable Logic Controller), 감지기(Sensor) 등과 같이 물리적인 형태로 존재하며 자체 저장 공간이 있고 응용 프로그램이 실행될 수 있는 환경을 제공한다.

② 응용 프로그램(Application)

특정한 목적을 수행하기 위하여 장치에 종속된 바코드 스캐너, 시리얼 포트, 기타 장치 등과 같은 디바이스들을 제어하면서 작업수행의 결과를 장치의 특정 위치에 기록하거나 통신을 통해 결과를 타 시스템에 서비스하는 역할의 실제적인 주체이다.

③ 데이터(Data)

장치에서 실행되는 응용 프로그램에서 생성되는 다양한 포맷의 결과물이다. 이것들은 장치의 디스크, 메모리 또는 응용 프로그램 내 버퍼, 파일, 데이터베이스 등의 다양한 형태로 저장된다. 또한 저장되지 않고 응용 프로그램과 연결된 타 시스템의 요청에 파라미터 형식으로 서비스되기도 한다.

3.2 데이터의 분류

응용 프로그램에서 생성된 데이터에 액세스하기 위해서는 장치의 저장위치 또는 데이터의 형태에 따라 세 가지로 분류 할 수 있으며, 분류된 유형에 따라 변화를 감시하는 방법이 달라진다.

① 파일 형태(File)

파일이란 컴퓨터에서 의미 있는 정보를 담은 텍스트 파일 또는 이진 파일을 말한다. 텍스트 파일은 구분자에 의해 항목(칼럼 또는 필드)을 구분할 수 있는 CSV(Character Separated Values) 파일과 그렇지 않은 유형으로 구분할 수 있다.

CSV는 구분자(콤마, 탭, 반각스페이스 등)를 기준으로 항목을 구분하여 저장한 비교적 단순한 포맷의 텍스트 파일이다. 문서 처리기나 편집기에서 읽거나 편집할 수 있으며, 많은 어플리케이션에서 취급하는 범용 형식이고 애플리케이션 간 데이터를 송수신 할 때에도 많이 사용되는 유형이다.

이진 형식의 파일은 컴퓨터에 저장과 특정한 처리 목적을 위해 이진 형식으로 인코딩된 데이터 파일이며 이미지, 사운드, 엑셀(Excel), 액세스(Access) 파일 등과 같은 다양한 유형이 있다.

② 데이터베이스 형태(Database)

커넥션 스트링(Connection String: 이하 연결자)으로 연결이 가능한 데이터베이스 유형이며, 특정 테이블들이 대상이 된다. 파일 유형으로 분류된 액세스(Access), 엑셀(Excel), CSV 파일 등도 연결자를 이용하여 데이터에 접근 가능하므로 효과적인 감시를 위해 데이터베이스 유형으로도 분류할 수 있다.

③ 파라미터 형태(Parameter)

데이터에 직접 접근할 수 없는 경우에 응용 프로그램에 의존하여 데이터를 파라미터로 전달되는 유형이다. 응용 프로그램의 제어 하에 있는 메모리 또는 버퍼의 기록된 데이터는 응용 프로그램간 통신을 이용한 요청

과 수신에 의해 텍스트, 2진, 8진 등의 다양한 형태로 서비스되는 유형이다.

4. 시스템 구현

4.1 시스템 요구사항 분석 및 시스템 환경

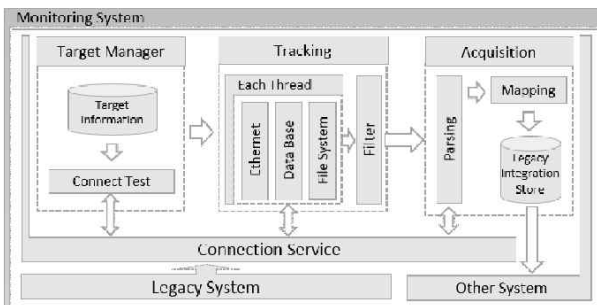
레거시 시스템의 각 장치들에 대한 연결과 각 데이터들의 감시 상태를 즉시 화면을 통해 모니터링 되어야 하고, 상위 또는 하위시스템과 독립적인 운영이 가능하며 타 시스템에 영향을 미쳐서는 안 된다. 이러한 독립적인 구성을 위해 다음과 같은 요구사항을 반영하였다.

- 타 시스템에 영향을 주지 않는 독립적인 운영을 위한 시스템 구조
- 기업의 규모를 고려한 저비용과 운영환경의 최적화
- 장치와 데이터의 속성 변경에 대한 확장성과 쉬운 유지보수

도출된 요구사항 분석에 의해 감시 시스템의 개발 및 운영환경은 Window Server 2010 운영체제를 사용하며 데이터 베이스는 MS-SQL Server 2012를 이용한다. 프로그래밍 .Net 3.5를 기반으로 C# 언어로 작성되었다. 개발은 레거시 시스템 데이터의 변화를 감시하는 인터페이스 모듈과 레거시 시스템의 상태 및 데이터를 제공하는 사용자 인터페이스 부분으로 나뉘었으며 총 6MM가 소요되었다.

4.2 시스템 구조

[Figure 3] 은 모니터링 시스템의 구조이며 레거시 시스템(Legacy System)과 감시시스템(Monitoring System)으로 구성되었다.



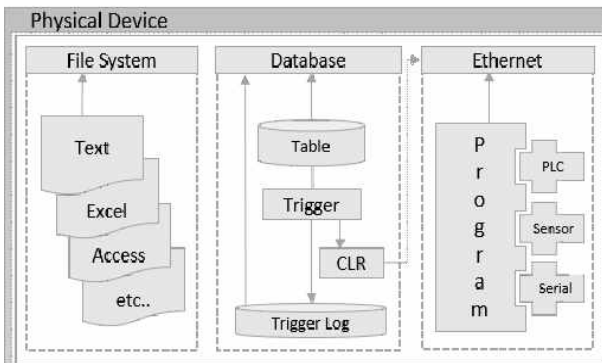
[Figure 3] System Architecture

레거시 시스템은 물리적인 장치, 응용 프로그램, 데이터로 구성되어 있고 이들은 장치의 파일 시스템, 데이터베이스 관리 시스템, 통신에 의존하여 연결되어 진다.

감시 시스템은 Target Manager(대상 관리자), Connection Service(연결 서비스), Tracking(추적) 등의 모듈로 구성된다. Target Manager는 물리적인 장치에 연결하기 위한 주소와 같은 기본 속성들과 파일 시스템 내에서 데이터의 논리적 위치, 데이터베이스의 대상 테이블과 연결자, 감시 데이터의 유효성 검사에 사용할 필터링에 대한 속성, 일괄 감시 방법을 사용하기 위한 주기설정 등의 정보를 유지 관리 하고 설정된 연결정보로 레거시 시스템에 연결을 테스트 할 수 있는 기능을 지원한다. Connection Service는 Target Manager에 등록된 연결정보를 이용하여 물리적 장치의 파일 시스템, 데이터베이스, 응용 프로그램과 연결 또는 통신을 지원하는 기능을 한다. Tracking은 Target Manager에 등록된 대상에 대한 정보를 이용하여 실시간 또는 설정된 주기정보를 이용하여 데이터의 변화를 감시하게 되며, 대상 데이터의 유형별로 각각의 스레드(Thread)에 할당되어 실행된다. Tracking에 의해 변화가 감지된 대상에 대한 정보는 등록된 대상과 일치 하는가 판단하기 위하여 Filter로 전달되어 유효성을 검사하고 정보는 수집 시스템으로 전달되어 Parsing과 Mapping을 거쳐 Legacy Integration Store(레거시 통합 데이터)에 저장된다. 이러한 감시 과정의 활동 상태는 유형별로 분류되어 실시간 모니터링이 가능하게 된다.

4.3 데이터 유형별 구현

데이터의 변화를 감시하기 위하여 데이터의 발생 빈도, 이용 목적, 실시간 감시 여부 등을 분석하고 중요도에 따라 각 유형별로 감시의 방법을 달리할 수 있다. 변화를 추적하기 위하여 데이터에 접근하기 위해서는 운영체제하의 파일 시스템에 의존하거나, 데이터베이스와 같은 관리 시스템에 의존한다. 그 외 버퍼, 메모리에 저장되거나 또는 저장되지 않는 실행 데이터들은 직접 액세스 할 수 없지만 서비스를 지원하는 응용 프로그램의 경우는 통신에 의존하는 방법이 사용될 수 있다. 의존 유형에 따라 세부 적용 방법을 사용할 수 있다. [Figure 4] 는 레거시 데이터의 의존 유형이다.



[Figure 4] Legacy data type

(1) 파일 시스템 의존형

파일 유형의 데이터는 상태의 변화가 생겼을 때 장치의 파일 시스템(File System)에 의존하여 실시간 알림을 받거나 액세스할 수 있다. 파일 시스템이란 운영체제가 파일을 발견 및 접근할 수 있도록 보관 또는 조직하는 관리체계이다. 네트워크 프로토콜을 이용하여 로컬 또는 원격 장치에 연결하고 파일 시스템에 접근하여 지정된 파일을 액세스 할 수 있다.

파일 유형의 데이터는 장치의 파일 시스템에 디렉터리와 파일로 생성되며, 디렉터리 및 파일의 이름은 날짜, 시간, 로트 번호, 장비의 속성, 검사기기 속성, 위치 등으로 다양하게 조합되어 명명된다.

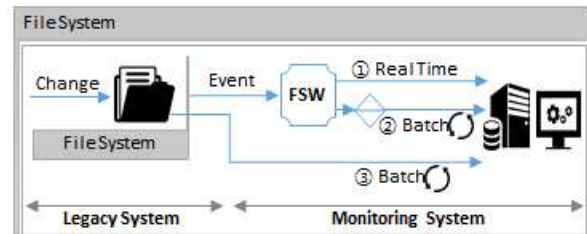
파일의 상태가 변경되었을 때 실시간으로 알림을 받기 위해 .NET Framework의 I/O 컴포넌트인 FSW(File System Watcher)를 사용하였다. FSW는 로컬 컴퓨터, 네트워크 드라이브, 또는 원격 컴퓨터에 있는 파일 시스템의 변경 알림을 수신하면서 디렉터리 또는 파일이 생성되거나 변경, 삭제될 때 실시간으로 이벤트를 발생시켜주는 기능을 한다.

<Table 1> Key properties of the FSW

Property	Function
Path	Gets or sets the path of the directory to watch.
Filter	Gets or sets the filter string used to determine what files are monitored in a directory.
IncludeSub directories	Gets or sets a value indicating whether subdirectories within the specified path should be monitored.

<Table 1> 은 FSW의 세 가지 주요 기본 속성이며, Filter 속성에서 정규표현 식을 지원하지 않아 다양한 이름과 형식으로 생성되는 디렉터리 및 파일을 필터링하기에는 제한적이어서 정규표현 식 (Regular

Expression)을 설정할 수 있는 속성을 추가 하였다. 추가 설정된 정규표현 식은 FSW로부터 수신된 이벤트의 속성 중에 대상의 경로를 식별하는 FullPath의 유효성을 판단하는 데에 사용되었다.



[Figure 5] File System Dependent Type Flow

레거시 데이터의 발생빈도는 시, 분, 초에 따라 한 건 또는 그 이상의 빅데이터가 실시간으로 발생하는 것이 일반적이며, 데이터의 중요도, 실시간 감시여부, 데이터 이용의 목적 등을 세밀히 분석하여 상황에 맞게 아래의 방법들을 적절히 적용할 수 있다.

① 이벤트 실시간 감시

FSW의 알림 이벤트를 실시간으로 전달 받아 감시하는 방법이다. 이벤트의 발생 시점은 파일의 생성, 변경, 삭제, 오류 등의 상태가 변경될 때 파일의 경로 등이 포함된 정보가 즉시 전달된다. 빅데이터 발생의 경우에는 많은 양의 이벤트가 전달되어 즉시 처리하지 못하고 누적되어 밀리는 현상이 발생하고, 시스템에 상당한 부하를 줄 수 있어 ②, ③의 방법들을 적용하는 것이 유리하다.

② 이벤트 일괄 감시

이벤트 실시간 감시 방법을 보완하여 FSW의 알림 이벤트를 즉시 처리하지 않고 이벤트의 파일 경로를 분석하여 중복을 배제하고 일정 시간 동안 임시 버퍼 또는 큐에 누적한다. 누적된 이벤트들은 대상 정보에 설정된 시간 또는 일정 주기에 일괄 처리하는 방식이다. 이 방식 또한 빅데이터의 경우 설정된 시간이 짧으면 효율성이 떨어질 수 있다.

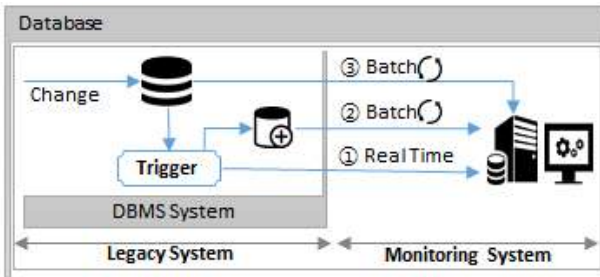
③ 주기적 일괄 감시

FSW를 사용하지 않고 실시간 감시가 필요치 않은 경우에 주기적으로 지정된 디렉터리의 정보를 모두 읽어 일괄처리 하는 방식이다. 이 방식은 기 조사된 파일들의 정보와 비교하여 변경된 파일을 조사하는 방법이다.

(2) 데이터베이스 의존형

데이터베이스는 공급자 별로 다양하고 또한 연결도

방식도 다양하지만 본 논문에서는 MSSQL 데이터베이스와 연결 방식은 ADO.NET을 사용하였다. 데이터베이스 테이블의 변경을 감시하는 방법에는 두 가지의 방법을 사용한다. 테이블에 행이 삽입, 변경, 삭제될 때 트리거를 이용하여 변화를 조사하는 방법과 일괄로 대상 테이블을 읽어 변경 여부를 조사하는 방법이다. 트리거를 사용하는 방법은 효과적이지만, 레거시 데이터베이스 관리 시스템(DBMS)에 트리거를 생성해야 하고 그에 필요한 권한 등을 지정해야 하는 등의 추가적인 설정이 필요하다. 또한 시스템의 이상 발생 시 그 원인의 대한 소제가 불분명한 경우에 분란의 소지가 생길 수 있다.



[Figure 6] Database Dependent Type Flow

① 트리거를 이용한 실시간 감시

장치의 DBMS가 MSSQL을 사용하는 경우, 트리거의 확장인 .NET CLR Assembly을 이용하여 데이터의 변화를 감시 시스템에 실시간 알람을 줄 수 있다. 이것은 C# 등의 언어로 작성된 프로그램으로 트리거가 발생하면 호출되어 실행되는 것이다. 트리거 발생 시 실행할 프로그램을 작성하고 컴파일 하여 DBMS에 트리거를 생성할 때 확장 옵션에 컴파일 된 CLR Assembly를 지정한다. CLR Assembly는 트리거로 전달된 매개변수를 받아 지정된 감시 시스템에 매개변수를 전달하고, 변경 알람을 해주는 간단한 구조의 통신 프로그램이다.

② 트리거를 이용한 일괄 감시

레거시 DBMS에 트리거와 감시용 Trigger Log 테이블을 생성하고, 트리거 발생 시 행의 변경 내용을 Trigger Log 테이블에 이력을 기록한다. 주기적으로 Trigger Log 테이블에 기록된 이력 데이터를 조사하는 방식이다.

③ 주기적 일괄 감시

트리거를 사용하지 않고 주기적으로 레거시 원본 데이터를 직접 조사하는 방법이다. 이 방법은 간단하고 쉬

운 반면에 레거시 테이블의 구성에 따라 변화를 조사하는 방법을 달리 할 수 있다.

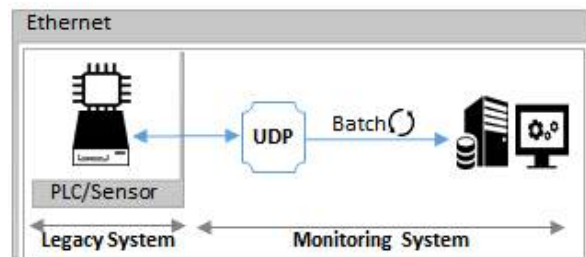
데이터 발생 시 테이블에 행이 누적(Append)만 되는 경우에는 테이블 행의 건수를 확인하는 것만으로 변화에 대한 인식이 가능하다. 삽입, 변경된 시간(Time Stamp)이 행에 존재한다면 그 시간 정보로도 인식 가능하고, 식별 가능한 고유한 주기가 순차적으로 증가한다면 최종 삽입된 순차번호로도 인식 가능하다. 순차적이지 않지만 고유한 주기가 존재하는 경우에는 기존 감시된 데이터와 조인함으로써 변화를 인식할 수 있다. 하지만 위에 열거된 경우에 해당되지 않는다면 처리된 데이터와 전체 풀 스캔 조인하여 조사함으로써 시스템에 상당한 부하를 줄 수 있다.

(3) 통신 의존형

제어기(PLC, Sensor 등)와 같은 장치는 응용 프로그램에 의해 발생된 데이터를 자체 메모리 또는 내부 버퍼에 기록하는 것이 일반적이다. 변경된 이력을 가지고 있는 경우도 있고, Sensor와 같은 감지기는 최종 카운트 된 결과만을 기록하기도 한다.

또한 통신이 가능한 것과 그렇지 않은 것으로 나눌 수 있으며, 본 논문에서는 통신이 가능한 장치로 제한하였다. PLC의 IEC61131-3과 같은 표준이 있지만 오래된 레거시 시스템의 경우는 이 표준이 준수되지 않은 경우가 많고, 문서화가 부실하거나, 공급자의 파산 등으로 유지보수 할 수 없는 등의 원인으로 많은 시간과 노력이 필요한 부분이기도 하다.

응용 프로그램에서 서비스 받기 위하여 데이터를 요청하는 방법도 장치의 공급자 별로 다양하고, 서비스된 데이터의 포맷도 제 각각 다른 비정형 포맷을 사용하기도 한다. 본 논문에서는 응용 프로그램에서 서비스 받기 위하여 데이터를 요청하고 수신하는 부분은 사용자 정의 프로시저(UDP:User Define Procedure)를 작성하여 사용하였다.

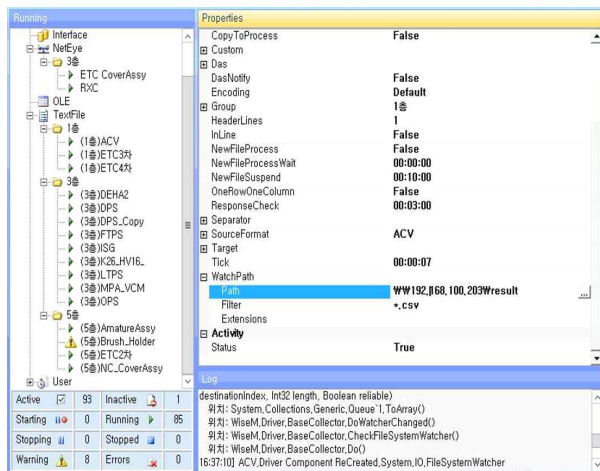


[Figure 7] Ethernet Dependent Type Flow

5. 시스템 적용

본 시스템은 ERP와 MES를 운영 중인 다양한 업체에 다수 적용하였다. [Figure 8] 는 93개의 장치 또는 데이터를 감시하는 모니터링 시스템이 구현된 화면 모습이다. 그림에 적용된 회사는 자동차 부품을 생산하는 업체이며 각 공정에서 생산된 반제품과 완제품의 자동 검사장비가 운영 중이었고, 그 검사 결과들은 주로 CSV 파일로 저장되고 있었다. CSV파일은 제품출하 시 출력하거나 파일로 첨부해야 했으나 실시간 데이터 수집의 어려움으로 인하여 수동으로 파일을 복사하여 엑셀파일로 변환하고 편집하는 등의 방식으로 수작업에 의존하고 있었다.

본 시스템을 적용하기 위해 CSV 파일을 분석한 결과 특정 항목으로 정상품과 불량품을 식별할 수 있었고, 식별된 결과는 실시간으로 수집되어 공정라인의 생산량으로 집계 가능한 상태였으며 각 검사 항목들은 통합 데이터베이스화가 가능하였다. 그 외 주요 공정라인의 생산량을 실시간 집계하기 위하여 감지장비 등이 추가 설치되었다.



[Figure 8] Monitoring screen

각 공정에 운영 중인 검사장비에 대한 연결정보와 논리적인 위치에 대한 속성 정보를 Target Manager 인터페이스를 통해 정의하였고, 그 외 센서와 같은 감지기에 대한 데이터 요청과 수집 등 사용자 모듈이 필요한 곳에는 사용자 인터페이스 모듈을 상속하여 처리로직을 추가 구현하였다. 변화가 감지된 데이터에 대한 정보는 유효성을 검사하기 위해 필터링 하여 수집 시스템 (Acquisition System)에 실시간으로 정보가 전달되었다. 수집 시스템은 전달 받은 데이터를 기초로 하여 Target Manager 에 등록된 Parsing과 Mapping정보를 이용하여 Legacy Integration Store 에 저장되었다.

The image shows a table with columns: SNo, ActDate, ActDate, Part_No, UMA_TP51, UMA_TP52, OMA_TP51, OMA_TP52, SNo. The table contains multiple rows of data, including dates like 2015.12.07 and values for Part_No, UMA_TP51, UMA_TP52, OMA_TP51, and OMA_TP52.

[Figure 9] Real-time query of the data

[Figure 9] 는 수집 시스템에 의해 저장된 통합 데이터베이스를 이용하여 MES에서 검사정보를 공정별, 작업오더 등으로 분류되어 실시간 조회하는 모습이다. 수집 시스템에서 제공된 Legacy Integration Store를 MES에 제공하여 주요 공정의 실시간 실적과 생산량 집계가 가능해졌으며 모니터링을 통하여 장치의 상태를 실시간으로 모니터링할 수 있었다. 고객으로부터 요청된 검사결과에 대한 정보를 즉시 제공하여 신뢰를 줄 수 있었으며, 또한 로트 또는 파트 번호 등의 특정 키워드에 대한 검사정보를 추적할 수 있었다.

6. 결론

본 논문에서는 레거시 시스템의 장치와 데이터를 유형별로 분류하고, 장치에서 발생된 데이터의 변화를 감시하는 접근방법에 대하여 정리하였으며, 실시간으로 장치의 활동 상태 및 데이터의 변화를 감시할 수 있는 모니터링 시스템을 구현하였다.

또한 원활한 유지보수를 위하여 전문적 또는 특수한 기술을 배제하고, .Net C#언어의 기본적인 기능을 최대한 활용하였고, 장치와 데이터의 추가 또는 변경에 유연하게 대처하기 위해 직관적인 사용자 인터페이스를 적용하여 새로운 장치가 추가되거나 데이터의 추가, 변경 등에 전문가의 도움 없이도 운영될 수 있게 구현하였다.

감시기 컴퓨터는 기업의 규모를 고려하여 저 사양의 컴퓨터로도 운영될 수 있도록 최소의 자원을 사용하도록 설계하였으며, 감시 시스템 운영에 필요한 대상에 대한 기본정보와 프로세스는 각기 다른 컴퓨터에 분산 저장하거나 실행될 수 있게 설계하여 이상 발생 시 빠른 조치가 이루어질 수 있게 하였다.

따라서 본 연구에서 개발된 실시간 모니터링시스템을 사용하면 분산되어 운영 중인 레거시 시스템의 정형 또는 비정형 데이터 변화의 감시가 필요한 다양한 분야에서 효율적으로 활용할 수 있을 것이다.

7. References

- [1] William M.Ulrich, (2002), "Legacy Systems: Transformation Strategies", Prentice Hall
- [2] Robert C.Seacord, (2003), "Modernizing Legacy Systems: Software Technologies, Engineering Processes, and Business Practices", Addison-Wesley Professional
- [3] J.M.Hughes, (2010), "Real World Instrumentation with Python: Automated Data Acquisition and Control Systems", O' Reilly Media
- [4] JeongHwa Park, EuiJong Lee, DooKwon Baik, (2015), "A Method for Data Collection and Allocation using Multi-thread in Relational Database System", Korea Information Science Society, 179-181
- [5] Sangrak Kim, Gilsang Jang, Chiwoon Cho, (2014), "Case Study of Design and Implementation for Hadoop-Based Integrated Facility Monitoring System", Journal of the Korean Institute of Industrial Engineers, 40(1):34-42
- [6] Tae Won Kyung, Sang Kuk Kim, (2005), "Web-based Data Integration System Implementation for Reliability Improvement of a Product", Information System Review, 7(2):117-128
- [7] Jai-Kyung Lee, Seung Woo Lee, So Jeong Nam and Jong Kweon Park, (2011), "Design of Information Acquisition System for Equipments on Ship Floor", The Korean Society of Mechanical Engineers, 35(1): 39-45
- [8] Seung Woo Lee, Jai-Kyung Lee, So Jung Nam, Jong Kweon Park, (2011), "Application of Data Acquisition System for MES", The Korean Society of Mechanical Engineers, 35(9):1063-1070
- [9] So Jeong Nam, Jai-Kyung Lee, Sung Woo Lee, Jong Kweon Park, (2010), "Acquisition of Data of Equipments on Shop Floor Using Interface Between Various Equipments", The Korean Society of Mechanical Engineers, 35(2):149-156

저자 소개

이재호



인천대학교 산업공학과 석사
현재 인천대학교 산업경영공학과
박사과정
관심분야: 물류정보시스템, 기업
정보시스템, 생산정보시스템
주소: 인천광역시 연수구 송도동
인천대학교 공과대학 산업경영공
학과

남호기



한양대학교 산업공학과 학사
Polytechnic Univ. 공학석사, 박사
현재 인천대학교 산업경영공학과
교수로 재직 중
관심분야: ERP, SCM, 물류정보
시스템, 생산관리
주소: 인천광역시 연수구 송도동 인
천대학교 공과대학 산업경영공학과

유우식



서울대학교 산업공학과 학사
한국과학기술원 산업공학과 석
사, 박사
현재 인천대학교 산업경영공학과
교수로 재직 중
관심분야: 물류정보시스템
CAD/CAM, 생산정보시스템
주소: 인천광역시 연수구 송도동
인천대학교 공과대학 산업경영공학과