

스마트공장 구축을 위한 데이터 인터페이스 미들웨어 개발

A Development of Data Interface Middleware for Building Smart Factory

정흥진¹, 송기형¹, 김보현¹,#
Hong Jin Jeong¹, Ki Hyeong Song¹, and Bo Hyun Kim¹,#

¹ 한국생산기술연구원 공정플랫폼연구부 (Department of Manufacturing Process Platform R&D, Korea Institute of Industrial Technology)
Corresponding Author / E-mail: bhkimr@kitech.re.kr, TEL: +82-31-8040-6174
ORCID: 0000-0002-7175-5473

KEYWORDS: Middleware (미들웨어), Data interface (데이터 연계), Smart factory (스마트공장), Information system interface (정보시스템 연계)

SMEs encounter many difficulties in integrating and operating various information systems introduced to build smart factories. The source of this difficulty comes from integrated management of data interface between information systems. This research proposes a data interface middleware that can operate and manage various data interfaces between information systems in an integrated manner. First, this study identifies the types of information systems and operational information needed to build smart factories and analyzes the ways of data interface and requirements suitable for the manufacturing environment of SMEs. Structure and detailed functions of the data interface middleware are designed based on the analysis results. The proposed data interface middleware consists of the function layer, engine layer, and DB layer. The function layer is a set of functions for operating the middleware, and the engine layer comprises core engines for executing the functions. The DB layer manages all information that gathers when the data interface is executed. We applied the proposed middleware to connect data between the existing ERP and newly introduced smart factory package software in SMEs. Application results show that the associated data types are consistent in the two systems, and accuracy of the data parsing process is reliable.

Manuscript received: June 22, 2021 / Revised: September 19, 2021 / Accepted: October 12, 2021

1. 서론

국내 중소 제조기업들은 자사의 경쟁력 강화 필수 수단으로 스마트공장 구축을 고려하고 있다. 따라서 스마트공장 구축은 Fig. 1에 나타난 것처럼 지속적으로 증가하는 추세이다.¹ 스마트공장 구축을 위해서 중소 제조기업이 가장 일반적으로 선택하는 방법은 정보시스템을 도입하여 운영하는 것이다. 정보시스템은 제조 환경에서 업무 처리나 경영의 의사결정에 필요한 데이터를 수집하고 처리하는 시스템이다.² 이러한 정보시스템은 제조 활동의 관리 범위와 역할에 따라 다양하게 정의되고 있는데, 대표적으로 전사적자원관리(Enterprise Resource Planning, ERP), 사전 생산계획 및 일정계획(Advanced Planning & Schedule, APS), 제조실행시스템(Manufacturing Execution System, MES), 품질관리시스템(Quality Management System, QMS) 등이 있다.

스마트공장은 여러 개의 다양한 정보시스템들을 필요로 하는데, 이때 가장 큰 이슈는 이러한 정보시스템들 사이의 빈번하게 주고받는 데이터들을 잘 관리하는 것이다.³ 이것은 정보시스템들의 운영 환경, 취급하는 데이터들의 종류, 속성과 정의 방식이 서로 다르기 때문이다. 즉, 다양한 정보시스템들 사이의 데이터 인터페이스(연계)를 통합적으로 운영 및 관리하는 것이 스마트공장의 운영 관리에서 매우 중요하고 필요하기 때문이다.

본 연구는 스마트공장에 필요한 다양한 정보시스템들 사이의 데이터 인터페이스를 통합적으로 운영·관리할 수 있는 미들웨어를 제안하고자 한다. 본 연구의 첫 번째 내용은 제조 활동의 관리 영역에 해당하는 정보시스템과 정보시스템들 사이의 데이터 관계를 분석하는 것이다. 두 번째 내용은 이러한 분석 결과를 바탕으로 데이터 인터페이스들을 통합적으로 관리하는 미들웨어의 구조 및 기능을 설계하고 구현하는 것이다. 마지막 내용은

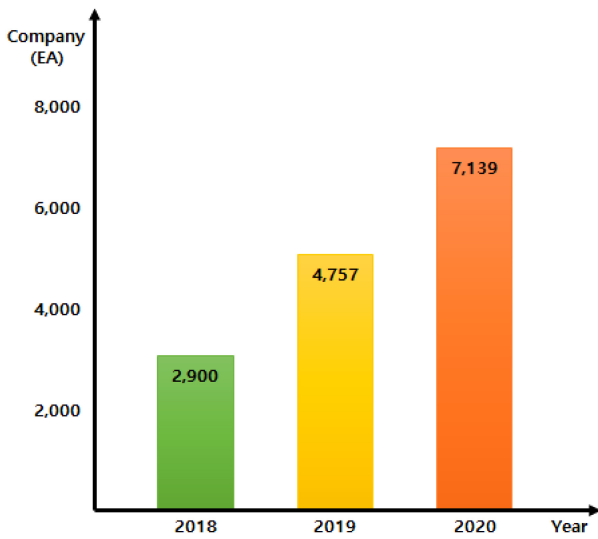


Fig. 1 Number of Small & medium size enterprise's smart factory

구현된 미들웨어를 실제 현장에 적용함으로써 제안된 미들웨어의 기능 및 효용성을 검증하고 하고자 한다.

2. 정보시스템과 데이터 인터페이스

2.1 스마트공장의 정보시스템

제조 활동은 관리 수준에 따라 크게 기업운영 및 관리(Enterprise Operation Management), 생산운영관리(Production Operation Management), 현장관리(Shop Control)의 세 가지 영역으로 구분하며, 각 영역에 적합한 정보시스템을 활용한다.^{4,5} 먼저 기업운영 및 관리 영역은 전반적인 기업의 운영 및 관리를

하는 업무 영역으로 활용되는 정보시스템은 전사적자원관리, 제품수명주기관리(Product Lifecycle Management, PLM), 공급망관리(Supply Chain Management, SCM) 등이 있다. 생산운영관리 영역은 제조 현장을 실시간으로 파악하고 업무를 운영하는 영역이며, 고급 생산계획 및 일정계획, 제조실행시스템, 품질관리시스템 등의 정보시스템을 활용한다. 현장관리 영역은 제조 설비·장비, 치공구 및 작업자 등을 운영하고 통제하며, 제조시점관리(Point of Production, POP), 산업용사물인터넷(Industrial Internet of Thing, IIoT), 인간기계인터페이스(Human Machine Interface, HMI), 센서 등의 정보시스템 및 디바이스가 활용된다.⁶⁻⁸

이러한 제조 활동의 관리 수준에 따라 활용되고 있는 정보시스템들은 데이터를 통하여 서로 유기적인 관계를 맺고 있으며, 일부 데이터들은 중복적으로 저장하거나 관리한다. 즉, 정보시스템들은 기업의 제조 활동을 지원하는데 있어서 다양한 데이터들을 서로 주고받으면서 유기적인 관계를 유지한다(Fig. 2 참조).

앞서 기술하였듯이 제조 활동 영역별 정보시스템들은 일반적으로 서로 다른 데이터를 저장하고 관리한다. 그렇지만 대부분의 정보시스템들은 필수적인 마스터 데이터(기본 정보)를 공통으로 가지고 있고, 지원하는 기능들이 겹치는 경우도 있기 때문에 데이터의 중복이 발생하게 된다. 여기서, 이러한 중복 데이터들은 정보시스템별로 다른 이름으로 저장 및 관리된다. 따라서 어떤 하나의 정보시스템에서 특정 데이터가 변경되면, 이것과 같은 데이터를 관리하는 정보시스템들에서도 변경사항이 실시간으로 반영되어야 한다. 그렇지 않으면, 이러한 데이터를 활용한 제조 업무의 의사결정은 필연적으로 오류가 발생하게 된다.

스마트공장의 성공적인 구축과 운영을 위한 정보시스템들 사이의 데이터 인터페이스는 지금까지 기술한 이유 때문에 반드시 오류 없이 정확하게 수행되어야 한다. 따라서 스마트공장 내부의 다양한 정보시스템들 사이의 데이터 인터페이스들을 통합

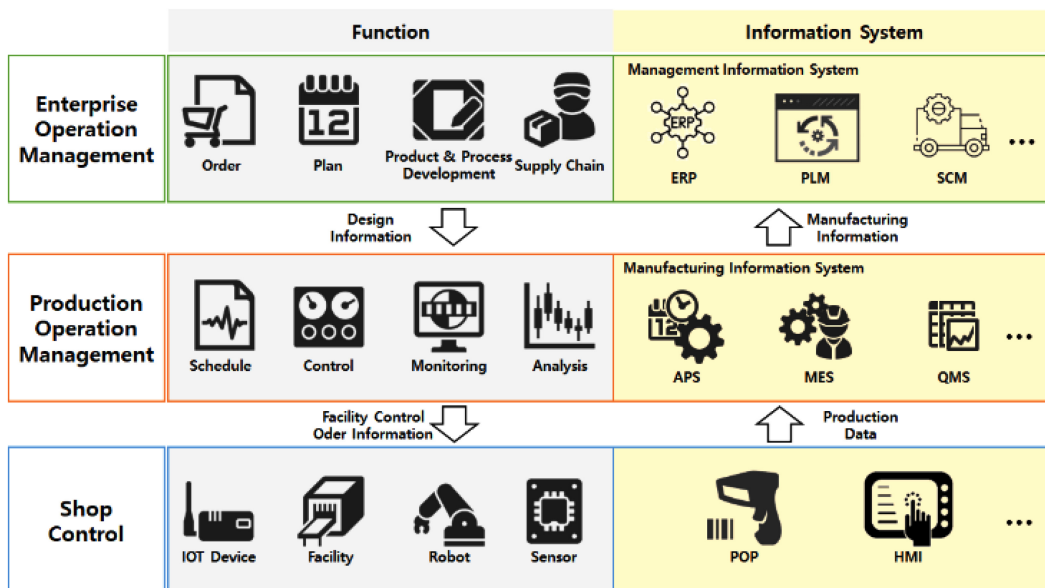


Fig. 2 Functions and information system in manufacturing activity areas

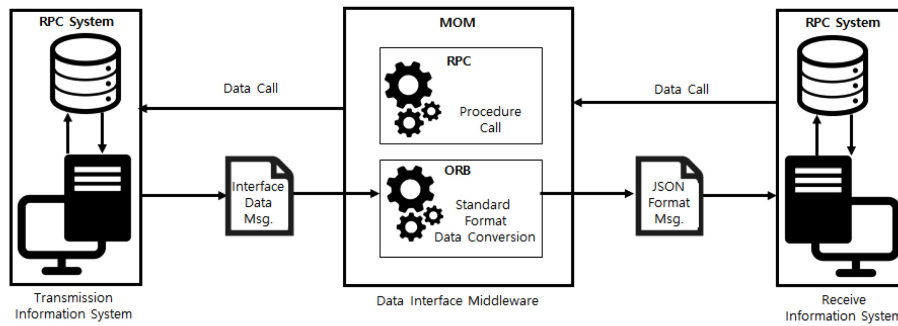


Fig. 3 Concept of data interface middleware

적이고, 효율적으로 수행하는 것이 본 연구에서 제안하는 데이터 인터페이스 미들웨어이다.

2.2 데이터 인터페이스 방식

제조기업의 정보시스템들은 서로 다른 운영환경에서 구동된다. 하나의 정보시스템과 다른 하나의 정보시스템 간의 데이터 연계를 위해서는 2개 정보시스템의 운영 환경이 고려된 데이터 인터페이스가 필요하다. 예를 들어, 스마트공장 내부에 N개의 정보시스템들이 있다면, 가능한 데이터 인터페이스 수는 N(N-1)/2 이다. 만약, 정보시스템 1개가 추가되면 새롭게 N개의 데이터 인터페이스가 추가되어야 하는데, 이것은 상당히 비효율적이고 관리의 어려움도 증가시킨다. 따라서 스마트공장의 성공적인 운영을 위해서는 정보시스템들 사이의 모든 조합에서도 데이터 인터페이스를 지원하고 쉽게 확장할 수 있는 데이터 인터페이스 미들웨어가 필요하다.

최근 들어 제조기업은 주요 정보시스템들의 핵심기능들만 추출하여 꾸러미로 만든 패키지형 정보시스템을 자사에 구축하여 활용하거나 클라우드 서비스로 활용하고 있다.⁹ 이러한 패키지형 정보시스템은 데이터를 내부적으로 통합 관리하기 때문에 별도의 데이터 인터페이스가 필요하지 않다. 그렇지만, 패키지형 정보시스템이 도입되더라도 대부분 제조기업은 기존의 정보시스템을 계속 운영하기를 희망하기 때문에 이들 사이의 인터페이스는 필수적으로 필요하다.

데이터 연계방식은 크게 직접 연계방식과 간접 연계방식으로 구분한다. 직접 연계방식은 연계 및 통합 구현이 단순 용이하고, 중간 매개체가 없어서 연계 처리의 성능이 비교적 좋지만, 시스템 간의 결합도가 높아서 시스템 변경에 민감한 단점이 있다. 간접 연계방식은 다양한 운영 환경의 연계 및 통합이 가능하고 보안이나 비즈니스 업무처리를 위한 로직을 자유롭게 반영하지만, 연계 구조와 메커니즘이 복잡하고 중간 매개체로 인한 성능저하 요소가 존재한다. 중소 제조 환경에서 이러한 정보시스템들은 제한적인 환경에서 운영되고, 데이터 연계를 쉽게 구현하기를 희망하기 때문에, 일반적으로 중소기업은 직접 연계방식의 데이터 인터페이스를 선호한다.

직접 연계방식은 대상 정보시스템의 송/수신 역할을 정의하고 대상 데이터를 직접 참조함으로써 데이터 인터페이스를 수행

한다.¹⁰ 대표적인 직접 연계 방식으로는 DB 링크, DB 연결, API (Application Programming Interface), JDBC (Java Database Connectivity), 하이퍼링크 등 다섯 가지가 있다.

첫째, DB 링크는 수신시스템의 DB에 송신시스템이 접근 가능한 DB 링크 객체를 생성하여 데이터를 연계하는 방식이다.¹¹ 둘째, DB 연결은 수신시스템의 WAS (Web Application Server)에서 송신시스템 DB로 연결하는 풀(Pool)을 참조하여 데이터를 연계하는 방식이다.¹² 셋째, API는 송신시스템의 DB에서 제공되는 데이터를 참조하여 데이터를 연계하는 방식이다.¹³ 넷째, JDBC는 수신시스템의 JDBC 드라이버를 이용하여 송신시스템의 DB와 연결하여 데이터를 연계하는 방식이다.¹⁴ 마지막으로 하이퍼링크는 웹 어플리케이션상에서 하이퍼링크 이용하여 데이터를 연계하는 방식이다.¹⁵

일반적으로 DB 연결과 API를 가장 많이 사용한다. DB 연결 방식은 송/수신 정보시스템의 DB에 접속하여 연계 대상 데이터를 직접 정의하는 방법이다. 이 방법은 보안에는 강하지만 연계 대상 데이터를 직접 정의하기 때문에 송/수신 정보시스템별로 별도의 추가 작업이 필요하다.^{16,17} API 방법은 송신 정보시스템에서 미리 정의한 데이터만 참조할 수 있어서 수신 정보시스템에서는 별도의 작업이 필요하지 않고, 새로운 정보시스템과의 데이터 인터페이스 추가가 비교적 용이하다.^{18,19}

3. 데이터 인터페이스 미들웨어 개발

제2장에서 기술한 데이터 연계 방식은 다중성과 보안성을 강조하였다. 따라서 본 연구는 다중성, 보안성을 바탕으로 다양성과, 확장성을 추가로 고려하여 데이터 인터페이스 미들웨어를 개발하고자 한다. 다중성은 미들웨어가 여러 개의 정보시스템 사이의 데이터 인터페이스를 동시에 수행하는 것을 의미하며, 보안성은 데이터 인터페이스를 수행하는 동안에 데이터의 보안이 유지되는 것을 의미한다. 그리고 본 연구에서 추가로 고려하는 다양성은 정보시스템이 활용하는 다양한 데이터관리시스템(DataBase Management System, DBMS) 간에도 데이터 인터페이스가 가능하는 것을 의미한다. 확장성은 정보시스템이 새로 도입되어도 데이터 인터페이스를 쉽게 추가 확장할 수 있는 것을

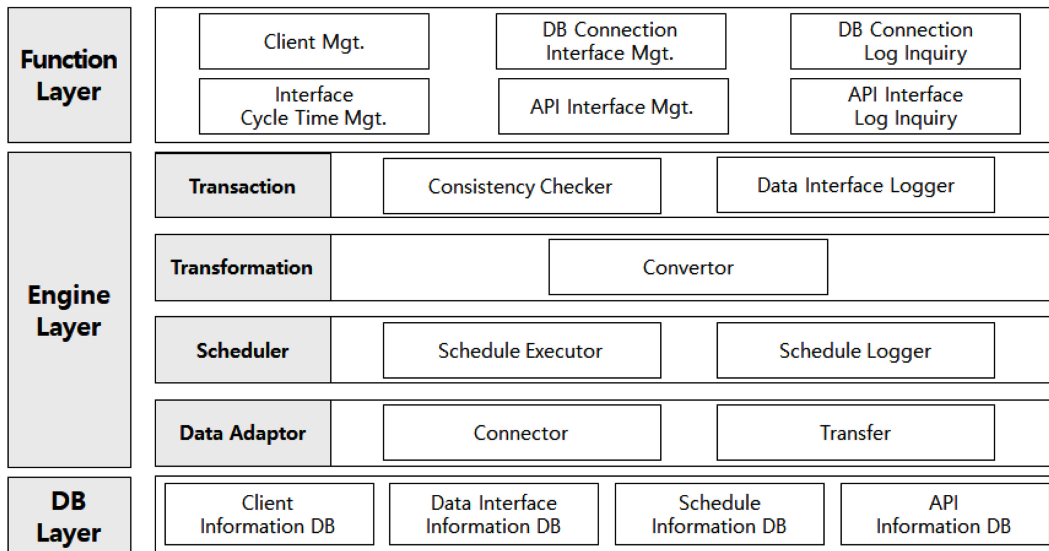


Fig. 4 Structure of data interface middleware

나타낸다.

국내 대부분의 중소 제조기업들은 데이터 보안을 위해서 모든 정보시스템 서버를 자사 내부에 놓고 운영하기를 원한다. 이와 같은 특수한 시스템 운영 환경 아래에서 데이터 인터페이스는 직접적인 DB 연결 방식이 바람직하다. 아울러, 그들은 기존의 레거시 시스템을 운영하고 있기 때문에 새로운 정보시스템이 도입되더라도 미들웨어는 데이터 인터페이스를 쉽게 추가할 수 있어야 한다.

이러한 고려사항을 반영한 미들웨어는 인터페이스의 대상 데이터를 호출하고 응답하는 메시지 전달 방식(Message-Oriented Middleware, MOM)으로 운영한다. 이때 데이터 호출 및 응답 방법은 원격 프로시저 호출(Remote Procedure Call, RPC)을 사용하며, 인터페이스의 대상 데이터는 표준 형식(Object Request Brokers, ORB)으로 변환하여 정보시스템에 전달된다(Fig. 3 참조).

3.1 데이터 인터페이스 미들웨어 설계

Fig. 4에 나타난 것처럼, 본 연구에서 제안하는 데이터 인터페이스 미들웨어는 기능 계층, 엔진 계층, DB 계층의 3개 계층으로 구성된다. 기능 계층은 미들웨어를 운영 및 관리하기 기능들의 집합이고, 엔진 계층은 기능 계층을 실행하기 위한 코어 엔진들로 구성된다. 그리고 DB 계층은 데이터 인터페이스를 실행하는데 있어서 필요한 정보를 관리하기 위한 것이다.

기능 계층은 클라이언트 관리, DB 연결 인터페이스 관리 및 로그 조회, API 인터페이스 관리 및 로그 조회, 인터페이스 주기 관리 기능들로 구성된다. 클라이언트 관리는 원격 프로시저 호출에 접근하기 위한 계정을 등록, 삭제하는 기능이다. DB 연결 인터페이스 관리는 정보시스템 DB의 데이터 구조를 확인하여 인터페이스 대상 데이터를 선택하고 규칙을 정의하는 기능이다. DB 연결 인터페이스 로그 조회는 등록된 DB 연결 인터페이스 인터페이스가 실행되면 수행된 결과를 기록하고 조회하는

기능이다. API 인터페이스 관리는 호출을 통해 데이터 인터페이스를 수행할 수 있도록 API를 등록 및 삭제하는 기능이다. API 인터페이스 로그 조회는 데이터 인터페이스가 실행될 때 API의 호출 결과를 기록하고 조회하는 기능이다. 인터페이스 주기 관리는 데이터 인터페이스별로 어느 시점에 얼마나 자주 인터페이스를 수행할지를 설정하는 기능이다.

엔진 계층은 데이터 인터페이스의 핵심 모듈로서 데이터 어댑터(Data Adaptor), 스케줄러(Scheduler), 트랜스포메이션(Transformation)과 트랜잭션(Transaction) 모듈로 구성된다. 데이터 어댑터는 DB 종류에 따라 별도로 구성되는데, DB 연결과 API 연결을 설정하는 커넥터와 해당 DB 내부의 대상 데이터를 전송하는 트랜스퍼로 구성된다. 스케줄러는 인터페이스 주기에서 설정된 시점과 주기에 따라 인터페이스를 실행하는 스케줄 실행기와 그 결과를 기록하고 조회하는 스케줄 로거로 구성된다. 트랜스포메이션은 데이터 인터페이스가 실행될 때 전송되는 데이터의 구조를 표준 형식(JavaScript Object Notation, JSON)으로 변환하는 컨버터가 있다. 트랜잭션은 데이터 인터페이스의 실행이 잘 수행되었는지 확인하는 데이터 인터페이스 검사기와 데이터 인터페이스 결과를 기록하는 데이터 인터페이스 로거로 구성된다.

DB 계층은 클라이언트 정보 DB, 데이터 인터페이스 정보 DB, 스케줄 정보 DB와 API 정보 DB로 구성된다. 클라이언트 정보 DB는 원격 프로시저 호출 연결 정보를 관리하고, 데이터 인터페이스 정보 DB는 DB 연결 인터페이스와 API 인터페이스 관리에서 설정한 정보를 관리한다. 그리고 스케줄 정보 DB는 데이터 인터페이스별 시점 및 주기 정보를 관리한다.

3.2 데이터 인터페이스 미들웨어 구현

데이터 인터페이스 미들웨어는 스마트공장 정보시스템의 다양한 운영 환경을 고려하여 두 가지 형태로 개발하였다. 그

(a) DB connection type

(b) API type

Fig. 5 User interface for client registration

하나는 중소 제조기업들이 가장 선호하는 사내에서만 운영하는 어플리케이션 형태이고, 나머지는 현재의 추세에 따라 클라우드 환경에서 운영하는 어플리케이션 형태이다. 기본적인 개발 환경은 리눅스 계열 운영체제인 CentOS이고, DBMS는 Window 서버와 MS-SQL 환경에서 개발하였다. 3.1절에서 설계된 6개의 주요기능, 4개의 핵심 모듈과 4개의 DB를 기반으로 미들웨어를 직접 운영 관리할 수 있도록 8개의 사용자 화면을 구현하였다.

여기서 클라이언트 등록, DB 연결 인터페이스 등록과 로그 기록 조회 등의 기능들을 중심으로 구현 내용을 설명하고자 한다. Fig. 6은 클라이언트에 대한 세부정보를 입력하여 클라이언트를 등록하는 사용자 화면이다. 클라이언트 ID는 미들웨어에서 자동으로 할당되고, 사용자는 클라이언트 이름을 별도로 입력하여 설정한다. 데이터 인터페이스 타입은 DB 연결 방식과 API 방식 중에서 하나를 선택하면 된다. Fig. 5(a)는 DB 연결

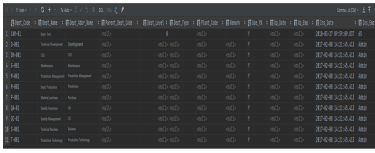
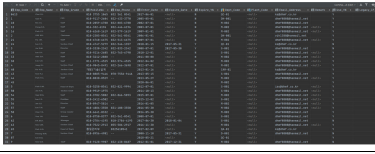
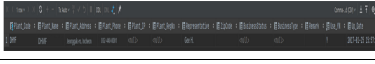
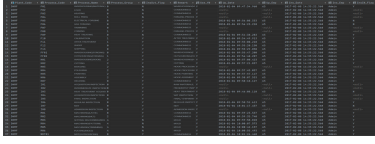
Fig. 6 User interface for DB to DB data interface registration

Fig. 7 User interface for DB to DB data interface log inquiry

방식을 설정하기 위하여 ‘DBMS’를 선택하면, 클라이언트 타입 정보를 입력하는 화면을 나타낸다. 여기서 IP, Port, Username, Password와 DB명을 입력하면 클라이언트 등록이 완료된다. 만약 API 방식을 설정하려면 ‘HTTP’를 선택하면 되는데, Fig. 5(b)에 나타난 것처럼 URL 필드에 API를 호출하는 사이트 정보를 입력하면 된다.

Fig. 6은 DB 연결 인터페이스 등록을 위한 사용자 화면인데, 여기서 ID는 미들웨어에서 자동으로 부여된다. 데이터 인터페이스 주기는 초, 분, 시간, 일, 달, 연의 필드값을 입력하면 설정된다. 예를 들어 초와 분의 필드값으로 10, 1을 나머지 필드를 *로 입력하면, 매 1분 30초마다 데이터 인터페이스를 수행하도록 설정한 것이다. 그리고 데이터 인터페이스 클라이언트는 데이터를 보내는 송신과 받는 수신 클라이언트를 지정하면 된다. 이렇게 설정된 인터페이스 정보를 현재 이용하지 않을 때는 No 항목을 체크하면 된다. 아래의 DB Schema 버튼을 누르면, 송/수신 클라이언트의 DB 스키마 정보를 확인할 수 있다. 여기서 DB 스키마 정보는 데이터 아답터를 통하여 가져오는데, 본 연구에서 현재 지원하는 데이터 아답터는 MS-SQL, My-SQL, MariaDB, Oracle 등이다. 오른쪽 Add 버튼을 누르면 Option과 Query 창이 추가된다. Option 부분에서는 Select One, Insert One, Update One, Delete One 중에서 하나를 선택하고, 대상이 되는 클라이언트

Table 1 List of master data

Mater data		List inquiry
Department	Dept_master (11 Rows)	
Employee	Emp_master (31 Rows)	
Plant	Plant_master (1 row)	
Process	Process_master (45 Rows)	

엔트 종류를 선택한다. Query 필드는 인터페이스 대상 데이터에 어떠한 규칙을 적용할지 코드 형태로 입력하는 것이다.

Fig. 7은 등록된 데이터 인터페이스 목록과 실행된 기록을 조회하는 사용자 화면이다. 여기서 사용자가 하나의 데이터 인터페이스를 특정하면, 이때 설정된 인터페이스 주기에 따라 인터페이스 실행 결과를 “완료(Complete)”와 “에러(Error)”로 표시한다. 이때 에러가 발생하면 인터페이스 시작 일시와 에러 발생 일시가 표시되고, 에러 메시지를 나타낸다. 에러 메시지를 선택하면 에러에 대한 자세한 내용이 팝업 창에 나타난다.

4. 적용사례

D사는 뿌리산업의 용접분야가 전문인 제조기업으로 주요 생산품은 건설기계용 중형 제판품이고, 2020년도 기준 매출 41억 원, 종업원 19명인 중소기업이다. 최근 들어, D사는 주요 고객인 건설기계 부품제조 기업들로부터 비용 절감과 경쟁력 강화에 대한 요구를 강하게 받고 있다. 이러한 이유로 D사는 스마트공장을 구축하기로 결정하였고, 뿌리산업 업종의 특성과 중소기업 제조 환경에 특화된 스마트공장 패키지 소프트웨어를 도입하였다. 여기서, 스마트공장 패키지 소프트웨어는 제조 활동에 필요한 주요 정보시스템들의 기본 기능들을 묶어서 템플릿으로 구성하고, 여기에 기업별로 요구하는 선택·특화 기능들을 패키징한 정보시스템이다. 이러한 패키지 소프트웨어는 대상 기업의 사내에서만 운영할 수 있고, 또한 클라우드 서비스 환경에서도 운영할 수 있다.²⁰

본 연구는 스마트공장 구축을 위해서 패키지 소프트웨어를 도입하려는 D사에 데이터 인터페이스 미들웨어를 적용하였다. 데이터 인터페이스의 적용 대상은 D사가 이미 보유 운영 중인 ERP 시스템과 새로 도입된 스마트공장 패키지 소프트웨어 사이의

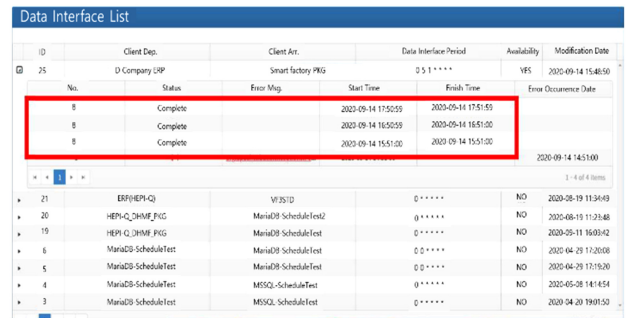


Fig. 8 Result of data interface

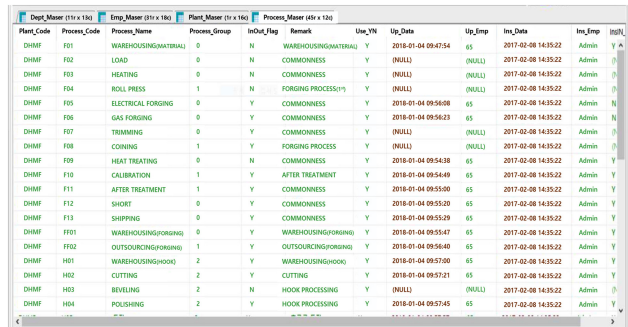


Fig. 9 Master data inquiry of smart factory PKG SW DB

데이터 연계이다. D사의 ERP 시스템은 Windows 계열의 운영체제와 MS-SQL를 사용하며, ERP 서버는 사내에 설치하여 운영하고 있다. 앞서 기술하였듯이, D사는 ERP 시스템의 운영 환경을 고려하여 도입된 스마트공장 패키지 소프트웨어도 사내에서만 운영하는 것을 요구하였기 때문에 미들웨어도 같은 환경에서 운영하도록 하였다. D사는 ERP 시스템에서 주로 회계업무 관련 기능만 사용하고 있어서, 스마트공장 패키지 소프트웨어와의 데이터 인터페이스 대상은 기준정보(Master Data)로 한정하였다. 대상 기준정보는 부서정보, 사원정보, 공장정보, 공정정보 등이며 Table 1에 상세하게 나타나 있다.

D사의 데이터 인터페이스는 DB 연결 방식으로 진행하였고, ERP 시스템과 스마트공장 패키지 소프트웨어의 DB 접속 권한은 미들웨어의 클라이언트 등록 기능을 통하여 등록하였다. 아울러 ERP를 송신시스템, 스마트공장 패키지 소프트웨어를 수신시스템으로 정의하였고, 데이터 인터페이스 주기를 설정하고 데이터 인터페이스 쿼리를 작성하였다. 데이터 인터페이스 주기는 테스트를 위해 1시간으로 설정하였고, 인터페이스의 실행 결과는 DB 연결 데이터 인터페이스 로그 조회 기능을 통하여 확인하였다. 데이터 인터페이스는 성공적으로 수행되었는데, 그 결과가 Fig. 8에 나타나 있다.

본 연구는 데이터 인터페이스가 정의한 규칙대로 실행되었는지 확인하기 위해서 추가적인 분석을 실시하였다. 분석 방법은 ERP DB와 스마트공장 패키지 소프트웨어 DB의 데이터 개수를 확인하고 데이터 처리 정확도를 확인하였다. 각 DB의 데이터 개수 확인은 D사 ERP DB에 입력한 기준정보 아이템 수와

스마트공장 패키지 소프트웨어 DB의 기준정보 아이템 수를 비교하는 것인데, 최종 확인 결과는 2개 시스템들의 기준정보 아이템 수가 모두 88개로 같았다. Fig. 9는 기준정보 아이템의 수신 여부를 확인하기 위해서 스마트공장 패키지 소프트웨어의 DB 테이블을 조회한 화면이다.

데이터 처리 정확도는 D사의 ERP에서 보낸 데이터 수와 스마트공장 패키지 소프트웨어가 받은 데이터 수를 비율로 확인하는 것인데, 식(1)과 같이 계산할 수 있다.

$$X = A/B \times 100 \tag{1}$$

여기서,

X: 데이터 처리 정확도(단위: %)

A: 스마트공장 패키지 소프트웨어에서 받은 데이터 수

B: D사의 ERP 시스템에서 보낸 데이터 수

데이터 처리 정확도를 확인하기 위해 1시간마다 실행되는 데이터 인터페이스 실행을 10회 반복한 결과로 확인하였다. D사의 ERP 시스템에서 보낸 누적 데이터 수는 880개이며, 스마트공장 패키지 소프트웨어에서 받은 누적 데이터 수는 880개로 데이터 처리 정확도는 100%로 확인하였다.

5. 결론

중소기업들은 스마트공장 구축을 위해서 도입한 정보시스템들을 통합적으로 운영하는 것에 있어서 가장 근본적이고 핵심적인 난제는 정보시스템들 사이의 데이터 연계이다. 이러한 난제를 극복하기 위해서 본 논문은 서로 다른 환경에서 운영되는 정보시스템들 사이의 체계적인 데이터 인터페이스 방법을 제안하였고, 이를 바탕으로 데이터 인터페이스 미들웨어를 설계하고 구현하였다. 그리고 구현한 미들웨어의 효용성을 살펴보기 위해서 실제 중소기업에서 운영 중인 ERP 시스템과 새롭게 도입된 스마트공장 패키지 소프트웨어의 데이터 인터페이스를 사례를 소개하였다.

본 연구는 스마트공장 구축에 필요한 정보시스템의 종류와 운영 정보를 분석하였고, 여기서 데이터의 통합관리 중요성과 데이터 인터페이스 미들웨어의 필요성을 확인하였다. 또한, 중소 제조 환경에서의 데이터 인터페이스는 직접 연계방식이 적합하다고 판단하였고, 데이터 인터페이스 미들웨어의 요구사항을 다중성, 보안성과 확장성으로 정리하였다. 데이터 인터페이스 방식과 요구사항을 바탕으로 본 연구는 데이터 인터페이스 미들웨어를 설계하고 구현하였다. 데이터 인터페이스 미들웨어는 크게 기능 계층, 엔진 계층, DB 계층으로 구성하였다. 기능 계층은 미들웨어를 운영 및 관리하기 6개의 기능으로 구성되며, 엔진 계층은 기능을 실행하기 위한 4개의 코어 엔진들로 구성된다. 그리고 DB 계층은 데이터 인터페이스를 실행하는데 필요한 모든 정보를 관리하는 것이다. 미들웨어 구조도를 기반으로

본 연구는 미들웨어를 사용하기 위한 8개의 사용자 화면을 구현하였다.

개발된 데이터 인터페이스 미들웨어는 중소 제조기업인 D사에 적용하였는데, 적용 대상은 D사가 운영 중인 ERP 시스템과 새롭게 도입한 스마트공장 패키지 소프트웨어 사이의 데이터 연계이다. 데이터 인터페이스 대상 데이터는 ERP 시스템의 기준정보이고, 데이터 인터페이스 방식은 DB 연결 방식으로 진행하였다. 데이터 인터페이스 결과는 송/수신 클라이언트의 DB를 조회하여 데이터 인터페이스 된 기준정보 아이템 수를 확인하고, 데이터 처리의 정확도를 확인하는 것이다. 최종 결과는 아이템 수가 일치하고, 데이터 누락 없이 100%의 처리 정확도로 나타났다. 그렇지만, 1개 사례의 수행 결과만으로 본 연구에서 제안한 데이터 인터페이스 미들웨어의 효용성을 주장하기엔 많이 부족하다. 앞으로 스마트공장 패키지 소프트웨어 도입하는 여러 기업의 다양한 운영환경에서 다양한 정보시스템들과의 데이터 인터페이스를 확대 적용함으로써 데이터 인터페이스 미들웨어의 효용성에 대한 추가적인 검증이 필요하다.

ACKNOWLEDGEMENT

이 연구는 2019년도 산업통상자원부 및 산업기술평가관리원 (KEIT) 연구비 지원에 의한 연구임(No. 20004577).

REFERENCES

1. Yum, J. S., "Status of Smart Factory in Korea, Press Release," Ministry of SMEs and Startups, 2021.
2. Kwon, S.-I. and Yang, J.-G., "A Study on the Structural Relationship among Technological Determinants, Manufacturing Operations, and Performances for Implementing a Smart Factory in Small Businesses," Journal of the Korea Academia-Industrial Cooperation Society, Vol. 21, No. 11, pp. 650-661, 2020.
3. Mittal, S., Khan, M. A., Purohit, J. K., Menon, K., Romero, D., et al., "A Smart Manufacturing Adoption Framework for SMEs," International Journal of Production Research, Vol. 58, No. 5, pp. 1555-1573, 2020.
4. IRSGlobal, "Manufacturing Innovation of Industry 4.0, Market·Technology·Standard·Analysis and Strategy in the Smart Manufacturing and Smart Factory," Information Research Service Global, 2020.
5. Jung, W.-K., Kim, D.-R., Lee, H., Lee, T.-H., Yang, I., et al., "Appropriate Smart Factory for SMEs: Concept, Application and Perspective," International Journal of Precision Engineering and Manufacturing, Vol. 22, No. 1, pp. 201-215, 2021.
6. Bae, K. H., Bae, S. J., Jeong, T. S., Heo, J. Y., Moon, S. K., et al., "Smart Factory Management & Technology," Dream Design,

2019.

7. Yoon, J. Y. and Kim, K. Y., "Status of Smart Manufacturing Industry and Advancement," Korea Institute for Industrial Economics & Trade, 2020.
8. Park, J.-S. and Kang, K.-S., "Strategies of Smart Factory Building and Application of Small & Medium-Sized Manufacturing Enterprises," Journal of the Korea Safety Management and Science, Vol. 19, No. 1, pp. 227-236, 2017.
9. Jeong, H. J., Kang, C. W., and Kim, B. H., "Development of a Quality Management System based on a Platform for Customized Application of Small & Medium Manufacturing Enterprises," Journal of the Korean Society for Precision Engineering, Vol. 35, No. 10, pp. 973-985, 2018.
10. Sunyaev, A., "Middleware," Internet Computing, pp. 125-154, 2020.
11. Zhuang, Y. and Yang, S., "Design of Enterprise Network Information System based on Industrial Interconnection," Data Processing Techniques and Applications for Cyber-Physical Systems (DPTA 2019), pp. 799-807, 2020.
12. Sud, K., "Database Connectivity," Practical Hapi, pp. 63-84, 2020.
13. Sunyaev, A., "Web Services," Internet Computing, pp. 155-194, 2020.
14. Sciore, E., "JDBC," Database Design and Implementation, pp. 15-47, 2020.
15. Ejiga, A. I. and Okeke, F., "Developing a Novel Approach of Data Storage and Retrieval for Online Multimedia GIS Applications Using Multimedia in GIS Approach," International Journal of Scientific Engineering and Science, Vol. 4, No. 5, pp. 71-76, 2020.
16. Blokdyk, G., "RDBMS Relational Database Management System a Complete Guide," 5starcooks, 2019.
17. Maheshwari, S. and Jain, R., "DBMS-Complete Practical Approach," Firewall Media, 2005.
18. Cleveland, S. B., Jamthe, A., Padhy, S., Stubbs, J., Packard, M., et al., "Tapis API Development with Python: Best Practices in Scientific Rest API Implementation: Experience Implementing a Distributed Stream API," Proc. of the Practice and Experience in Advanced Research Computing Conference, pp. 181-187, 2020.
19. Choi, M., "A Performance Analysis of Restful Open API Information System," Proc. of the International Conference on Future Generation Information Technology, pp. 59-64, 2012.
20. Song, K. H., Jeong, H. J., Lee, D. Y., and Kim, B. H., "Package Software Configuration and Cloud-Based Service System for Building a Smart Factory in the Root Industry," Journal of the Korean Society of Manufacturing Technology Engineer, Vol. 29, No. 4, pp. 323-330, 2020.



Hong Jin Jeong

Researcher in Korea Institute of Industrial Technology. His research interest is smart manufacturing, manufacturing information system and quality management.

E-mail: jhj0706@kitech.re.kr



Ki Hyeong Song

Researcher in Korea Institute of Industrial Technology. His research interest is precision cutting process, sub-micro pattern cutting processing and process monitoring.

E-mail: skhgood@kitech.re.kr



Bo Hyun Kim

Principal Researcher in Korea Institute of Industrial Technology. His research interest is manufacturing information system, smart manufacturing and manufacturing service.

E-mail: bhkim@kitech.re.kr