

## 제조실행시스템에서의 BPEL 기반 워크플로우 관리시스템의 적용\*

박동진\*\* · 장병훈\*\*\*

### Implementation of BPEL based Workflow Management System in Manufacturing Execution Systems\*

Dongjin Park\*\* · Byoung-Hoon Jang\*\*\*

#### ■ Abstract ■

This paper outlines opportunities and challenges in the implementation of BPEL based WFMS(WorkFlow Management System) for the MES(Manufacturing Execution Systems) in semiconductor manufacturing. At present, the most MESs in semiconductor wafer fabrication shop have the problems in terms of application software integration, reactivity, and adaptability. When a plant has to produce new product mix, remodel the manufacturing execution process, or replace obsolete equipments, the principal road blocks for responding to new manufacturing environment are the difficulties in porting existing application software to new configurations. In this paper, the issues about WFMS technologies including BPEL standard applied for MES are presented. And then, we introduce the integrated development framework named nanoFlow which is optimized for developing the BPEL based WFMS application for automated manufacturing system. And we describe a WFMS implemented with using nanoFlow framework, review and evaluate the system.

Keyword : BPEL, workflow, MES, Lot tracking, Framework, XML

논문투고일 : 2009년 07월 16일      논문수정완료일 : 2009년 12월 11일      논문게재확정일 : 2009년 12월 14일

\* 이 논문은 2007년 공주대학교 학술연구지원사업의 연구비지원에 의하여 연구되었음.

\*\* 공주대학교 산업시스템공학과 교수

\*\*\* 에임시스템(주) R&D 센터장

# 1. 서론

현대의 제조시스템은 잦은 요구사항의 변화, 리드타임의 압력, 새로운 기기의 도입에 따른 호환성 등 수많은 문제에 직면하고 있다. 기업이 제조경쟁력을 확보하기 위해서는 제조의 동기화가 매우 중요하며, 이를 위해서 제조실행 시스템(MES : Manufacturing Execution System)에 많은 투자를 하고 있다. MES는 특히 소프트웨어 측면에서 제조환경의 변화에 대하여 유연하게 대처할 수 있어야 하며, 이질적인 컴포넌트들을 효과적으로 통합할 수 있는 능력을 확보하는 것이 매우 중요하다.

반도체 기업의 웨이퍼 제조시스템도 마찬가지다. 웨이퍼를 제조하기 위해서는 복잡한 개별 단계들을 거치는데 이 과정에서 소프트웨어의 중요성은 점점 증대되고 있다. 소프트웨어가 없이는 생산라인이 작동하지 않을 뿐 아니라, 소프트웨어 문제로 인해 생산의 지연은 물론 새로운 생산라인의 구축에 있어서 장애요인이 되기도 한다. 구체적으로, 첫째, 제조 현장의 소프트웨어 구성은 매우 이질적이다. 즉 많은 응용프로그램이 서로 다른 운영체제하에서, 다양한 데이터베이스를 이용하고, 다양한 프로토콜을 통하여 커뮤니케이션을 하기 때문에 시스템 통합(integration)을 어렵게 한다. 둘째, 제조현장의 생산공정순서는 상황에 따라 계속적으로 변화한다. 따라서 이러한 환경의 변화에 대하여 소프트웨어적으로 신속하게 대처하여야 하는데, 운영로직이 프로그램의 코드형태로 되어 있는 현실에서는 이에 대한 신속한 반응성(reactivity)이 결여 되어 있다. 셋째, 제조현장에서는 노후기기의 교체 및 새로운 툴의 도입이 계속적으로 이루어지는데, 새로운 생산환경에 대하여 소프트웨어적으로 유연하게 대처하는 적응성(adaptability)이 필요하다.

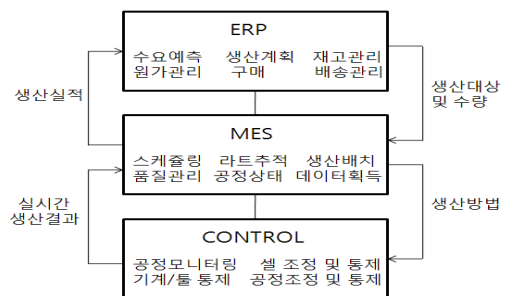
본 연구는 워크플로우(workflow) 기술을 적용하여, 이상에서 언급된 반도체 제조공정을 위한 MES 수준에서의 소프트웨어 이슈들을 해결하는 방안을 제시한다. 구체적으로 제 2장에서는 MES 수준에

서 워크플로우 관리시스템(WFMS : Workflow Management System) 도입의 필요성을 주장하였으며, 워크플로우 표준 언어로서 BPEL(Business Process Execution Language)을 채택한 WFMS의 아키텍처를 기술한다. 제 3장에서는 반도체 제조시스템의 WFMS 솔루션인 nanoFlow를 소개하고 MES 주요 기능 중의 하나인 로트추적(lot tracking)을 구현하는 과정을 소개한다. 또한 실제 구현한 시스템을 통한 경험과 효과를 기술하고, 마지막으로 제 4장에서는 결론을 맺는다.

# 2. 연구의 배경

## 2.1 MES에서 워크플로우 기술의 도입

제조기업의 정보시스템은 제조현장에서부터 공장운영계획에 이르기까지 의사결정의 계층적 구조에 따라 ERP 수준, MES 수준, 그리고 Control 수준으로 구분된다[2]. 각 수준에는 다양한 기능들이 포함되어 있으며, 이들이 정보시스템으로 인하여 수평으로 그리고 수직적으로 서로 통합됨으로 인하여 제조활동을 위한 정보를 원활하게 제공하는 것이다. [그림 1]은 각 수준들의 관계와 상세기능들을 보여준다.



[그림 1] 제조기업의 정보체계 수준 및 기능

여기서 MES는 ERP로부터 몇 주 단위의 생산계획을 전달받으며, 생산이 진행되면서 생산실적을 ERP로 피드백 한다. 또한 MES는 제조현장에서의 자세한 생산정보를 다루며, 하위의 Control

수준에 있는 제조기기/툴에게 어떤 작업을 할 것인가에 대한 자세한 정보를 내려 보내기도 한다.

MES 솔루션 개발자와 컨설턴트들은 MES가 미래의 변화에 대응하고 유연하게 적응하기 위해서는 다음과 같은 요구사항을 만족시키는 소프트웨어 아키텍처를 가져야 한다고 주장한다[3]. 첫째, ERP 하단에 있는 모든 기능들을 수평적으로 통합할 수 있는 기반을 갖추어야 한다. 둘째, 표준을 채택하며 요구사항에 따라 확장이 가능한 구조여야 한다. 셋째, 프로세스 및 상세요구사항에 표준 모듈을 쉽게 적용가능한 구조여야 한다, 넷째, 모든 수준에 걸쳐 표준화된 인터페이스의 채택하여야 한다.

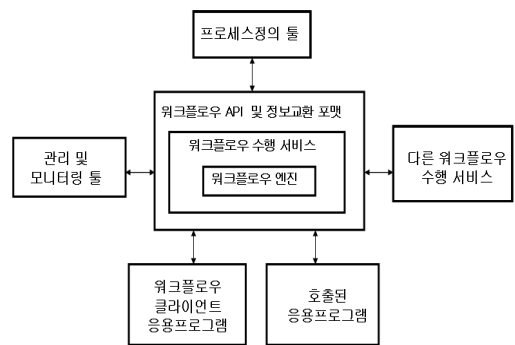
제조환경에서 WFMS를 포함한 워크플로우 기술은 주로 ERP 수준의 비즈니스 프로세스의 모델링, 리엔지니어링 그리고 정보화에 초점이 맞추어져 왔다[5, 9]. 그러나 WFMS의 다음과 같은 장점들로 인해서 MES에서 충분히 적용되어 질 수 있음을 알 수 있다. 첫째, 응용프로그램으로부터 비즈니스 로직 및 비즈니스 툴을 분리함으로써 업무가 변경되더라도 프로그램의 수정이 용이하다. 둘째, 비주얼한 프로세스 모델링 툴의 이용으로 최종사용자가 직접 비즈니스 로직을 관리할 수 있다. 셋째, 표준을 기반으로 한 시스템의 확장이 가능하다. 넷째, 각 컴포넌트들을 독립적으로 개발하고 재사용할 수 있다. 이러한 특성으로 인해서 제조기업의 MES 수준에도 WFMS가 적용되었다[1, 4].

이상에서 언급된 요구사항들을 만족시키는 소프트웨어 아키텍처를 선택하기 위해서는 기반기술 면에서 표준의 채택이 매우 중요하다. WfMC(Workflow Management Coalition)에서는 모든 워크플로우 솔루션 개발자가 지켜야할 5개의 카테고리에서 "Workflow Standards"를 제시한다. 이것은 결국 사용자 환경 및 소프트웨어 솔루션이 다를지라도 통합 및 인터페이스가 가능하게 한다. MES는 비즈니스 애플리케이션 중에서 가장 다양한 툴들이 서로 결합되어야 할 영역이므로 현재의 어떤 아키텍처 중 WFMS의 적용이 가장 바람직하다고

볼 수 있다.

## 2.2 반도체 제조시스템에서 WFMS의 적용

워크플로우 참조 모델은 WFMS의 기능적 구성요소들과 이들 간의 인터페이스를 포함한 WFMS의 본원적 구조를 보여준다. [그림 2]에서처럼 구성요소들은 프로세스 정의 툴, 워크플로우 엔진을 포함한 워크플로우 수행 서비스, 관리 및 모니터링 툴, 클라이언트 응용프로그램, 호출가능한 응용프로그램 등을 포함한다. 워크플로우 API(WAPI)와 상호교환포맷은 WfMC에서 제안한 것으로 WFMS의 서로 다른 구성요소들과 응용프로그램 간에 상호운용성(interoperability)을 확보하기 위한 명세서이다[8]. WfMC 참조모델에서는 5개의 구성요소들 간 인터페이스도 정의하고 있다.



[그림 2] WfMC 참조 모형

반도체 제조시스템에 있어서 제조프로세스는 매우 복잡하다. 웨이퍼를 생산하기 위해서는 수백가지의 세부단계를 거친다. 생산현장의 관리 프로세스들을 살펴보면 생산운영계획, 자원할당 및 스케줄링, 로트 추적, 제조공정의 명세 및 통제, 성과 모니터링, 현장 내 각종기기의 통제 및 모니터링 등이 있다. 이상의 각 프로세스들은 계층화된 세부 프로세스들로 구성되고 필요한 경우에 의사결정이 이루어져야 하며 자동으로 툴이 적용되기도 한다. 이상의 모든 프로세스에 WFMS가 적용됨으로써 반도체 제조시스템에서 소프트웨어의 이슈들

인 통합, 반응성 및 적응성을 확보할 수 있다. 아래 <표 1>은 반도체 제조시스템 WFMS에 특화된 기능들을 정리한 것이다.

<표 1> 반도체 제조분야의 WFMS 필요기능

분야	요구사항
기본 기능	병렬처리 멀티 트랜잭션 관리 짧은 주기의 트랜잭션 관리 데이터 변환 및 추출 동적 로딩 및 호출 조건을 통한 분기 처리 조건을 통한 반복 처리 에러 및 예외 관리
커뮤니케이션 패턴	(비)동기처리 추상화된 미들웨어 기능 전, 후 메시지 연관 기능 기간(Legacy) 시스템과 통합 기능 메시지 전달 및 처리 기능

### 2.3 BPEL 기반의 WFMS

WFMS 솔루션들은 비즈니스 프로세스를 모델링하고, 명세하고, 런타임 환경에서 실행하는데 있어서 다양한 기술들을 적용한다. 워크플로우의 모델링 및 실행방법은 크게 Choreography 언어와 Orchestration 언어에 의한 방법으로 구분된다. Choreography는 특정한 주도권을 가진 엔터티가 없으면서 비즈니스 엔터티들 간에 장기적이고 지속적인 상태(long-lived state)에서 서로 협조하면서 이벤트가 발생할 시 각자가 자기역할을 함으로써 프로세스를 처리하는 기술이다. 즉 peer-to-peer 방식으로 협업하는 모델이다. 반면에 Orchestration은 하나의 비즈니스 엔터티에 초점을 맞추는 것으로, 특정지점 한곳에서 사전에 정의된 대로 프로세스의 전체를 통제하는 것이다. 즉 비즈니스를 주도하는 하나의 참여자 관점에서 여러 프로세스들을 통합하는 모델이다[6].

MES 수준의 프로세스들은 첫째, 제조현장내부에서 제한된 수의 엔터티들이 일정한 순서에 의하여 통합되면서 프로세스들 간에 정보흐름이 이루어지는 것이다. 이러한 특성은 WFMS의 Orches-

tration 모델이 적합하다. 둘째, MES에서는 각 프로세스들 간에 수평적 통합(horizontal integration)을 하거나 상위시스템인 ERP와 하위시스템인 Control 수준의 프로세스들과의 수직적 통합(vertical integration)이 필요하므로 이질적인 프로세스 간에 통합이 가능한 표준 인터페이스가 필요하다[7]. 최근 WFMS 솔루션들은 내부적으로 프로세스를 표현하는 XML 기반의 프로세스 정의 언어를 채택하는데, 그 중에서 BPEL이 Orchestration 형식의 서비스를 지원할 뿐 아니라 표준 인터페이스를 제공함으로써 프로세스의 수평적 그리고 수직적 통합을 가능케 한다.

BPEL은 웹 서비스를 구성하여 비즈니스 프로세스의 행동을 명세하는 XML 기반의 언어이다. 하나의 프로세스는 파트너들과의 상호교류 관계로 정의된다. 파트너는 프로세스에게 서비스를 제공하거나 거꾸로 서비스를 요청하기도 한다. BPEL은 서비스를 의미가 있고 일정한 순서로 명세함으로써 웹 서비스를 Orchestrate 한다. 또한 XML 기반의 흐름 언어(flow language)이므로 if문, while문, 병렬수행 등과 같은 구조적 프로그래밍도 가능하다. [그림 3]은 BPEL 프로세스 정의의 간단한 예이다.

```

<process name = "BC01workFlow" ... properties ...>
  <partnerLinks> //파트너 정의(WSDL 파일로
    //부터 속성을 읽어온다).
    <partnerLink name = "client" ... properties .../>
  </partnerLinks>
  <variables> //변수 목록
    <variable name = "input" messageType
      = tns:BC01workFlowRequestMessage"/>
    <variable name = "output" messageType
      = tns:BC01workFlowResponseMessage"/>
  </variables>
  <!-- WorkFlow가 시작되는 부분-->
  <sequence name = "main">
    <receive name = "receiveInput" ... properties .../>
    <invoke name = "callbackClient"
      ... properties... />
    .....
  </sequence>
</process>

```

[그림 3] BPEL 프로세스의 예

BPEL 프로세스는 여러 개의 스텝으로 구성되며, 각 스텝은 <invoke>, <assign>, <reply>, <receive>, <throw>, <wait>, 그리고 <terminate>와 같은 기본 액티비티와 <sequence>, <flow>, <switch>, 그리고 <pick>과 같은 구조 액티비티로 구성된다. 또한 메시지는 기본적으로 복잡한 변수로서 표현된다. [그림 3]은 메시지를 받아서 웹 서비스를 호출하는 간단한 비즈니스 프로세스를 보여준다.

### 3. 반도체기업 MES에 WFMS의 적용

#### 3.1 워크플로우 기반 프레임워크

지금까지 반도체 제조시스템 MES의 개발 프레임워크는 소프트웨어 개발 측면에서 보면 주로 객체지향 프레임워크 하에서 분산처리방식인 CORBA를 채택하였다. 구체적으로 MES의 각 기능들을 독립적인 모듈로 구성하고 통합을 위해서 peer-to-peer 방식으로 인터페이스를 구성한다. 또한 기능이 추가되면 새로운 모듈과의 인터페이스를 개발하여 추가하는 방식을 택하였다. 이러한 방식은 다음과 같은 문제점들을 보인다. 첫째, 생산 운영로직이 코드 형태로 구현되어 개발자 개인에 대한 의존도가 높고 변경에 많은 시간이 소요되며 오류 발생 가능성이 높다. 둘째, 모듈간 인터페이스를 위한 메시지 전달 영역과 비즈니스 영역이 혼재되어 유지보수 및 관리가 어렵다. 셋째, 인터페이스를 위한 표준의 채택이 없어서 모듈의 재사용성은 물론 통합성 및 이식성이 부족하다.

이상과 같은 문제를 해결하는 방법은 새로운 개발 프레임워크의 도입이 필요하다. 즉 워크플로우 형태의 가시성을 확보하여 생산운영 로직의 분석을 용이하게 하고 개발생산성을 높이고 신속한 변경요구 대응과 오류발생 가능성을 최소화하는 프레임워크의 도입이 필요하다. 또한 프레임워크는 응용프로그램 통합을 위한 표준인터페이스를 제공하며 프로그램의 관리 및 모니터링이 가능하여야 한다.

#### 3.2 프레임워크의 구현

본 연구를 통하여 구현된 개발 프레임워크는 프로세스의 흐름을 “시각화” 시켜주는 워크플로우 기반의 개발환경을 제공한다. 구체적으로 자바나 닷넷을 위한 오픈 소스 애플리케이션 프레임워크인 스프링 프레임워크를 기반으로 하며, 서비스 바인딩(binding) 구조를 경량화하기 위해 BPEL에서 해당 bean을 직접 호출할 수 있도록 한다. 개발 패턴 관점에서는 오퍼레이션이 처리할 요소의 클래스를 변경하지 않고도 새로운 오퍼레이션을 정의할 수 있는 visitor 패턴을 사용하여 구조와 데이터 처리를 분리할 수 있게 한다. 또한 도메인 관점에서는 수년간 공장자동화 시스템을 개발, 적용함으로써 산출된 최소한의 기능들을 프레임워크 안에 포함한다. 즉 프레임워크 내에서 BPEL 표준에서 정의한 17개의 기본 액티비티와 공장자동화에 적합한 MES 액티비티를 visitor 패턴 형식으로 구현할 수 있도록 하였다.

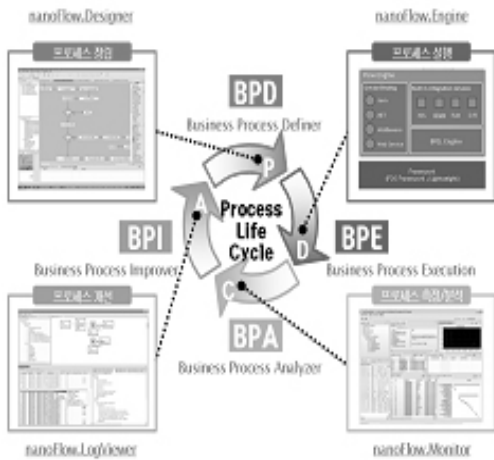
이러한 개념을 구체화한 것이 nanoFlow이다. 이는 범용 워크플로우 솔루션으로 공장용 응용시스템 개발에 최적화되어 있다. 솔루션은 [그림 4]와 같이 크게 개발환경(nanoFlow.Deginer), 운용환경(nanoFlow.Engine), 관리환경(nanoFlow.LogViewer), 그리고 모니터링환경(nanoFlow.Monitor)으로 구성된다. 각 구성요소들은 약결합(loosely coupled)되어 있어서 독자적인 운영이 가능하고 메시지 전달이나 원격프로시저 호출 등에 의해서 상호 연결 및 결합이 가능하다.

nanoFlow.Deginer는 Eclipse 환경의 플러그인으로 개발되었으며 드래그-드롭 방식으로 비즈니스 로직을 모델링 할 수 있다. 또한 서비스 리플렉션(reflection)을 통해 완성된 모델의 유효성 검사 가능하고 자바 클래스 서비스 컴파일과 동시에 모델링 할 수 있는 PartnerTreeView 가 제공되므로 실시간 테스트가 가능하다.

nanoFlow.Engine은 BPEL 파일을 로딩하여 실행하는 핵심 컴포넌트이다. 워크플로우 프로세스의

성공 여부, 총소요시간, 변동 등을 실시간으로 모니터링하고 워크플로우 엔진의 모든 상황을 원격으로 관리하는 JMX(Java Management eXtensions) 콘솔 기능을 포함한다. 성능측면에서 1초 당 100개 이상의 워크플로우 프로세스를 처리하므로 고속의 처리 속도와 안정성을 보장한다.

nanoFlow.Monitor는 nanoFlow 기반의 모든 응용 프로그램을 실시간으로 모니터링하고, 비즈니스의 로직을 추적하고, 결과와 통계치를 생성한다. nanoFlow.LogViewer는 애플리케이션이 생성하는 로그 파일을 기준으로 BPEL의 성공여부, 운용경과(elapsed) 시간에 대한 최대, 최소, 평균값을 다양한 형태로 보여줌으로써, 향후 발생될 문제를 사전에 예측하고 방지할 수 있다.



[그림 4] nanoFlow 아키텍처

3.3 로트 추적 프로세스의 적용

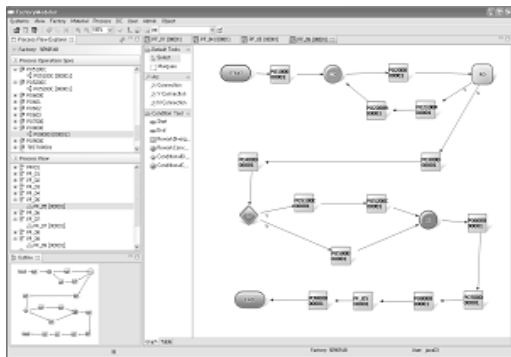
웨이퍼 가공은 매우 복잡한 공정을 거치며 각 제품품은 로트단위로 관리되어진다. MES의 로트추적 기능은 로트를 생성하고, 각 세부공정상에 있는 로트의 공정이력정보를 구축하고, 생산 공정상에 있는 각 로트들을 가시화하고 이들을 통제하는 것 등을 포함한다. 구체적으로 <표 2>와 같은 기능들이 포함한다.

<표 2> 로트 추적 상세 기능

생산 지시	작업지시	WO 시작
		WO 완료
기본기능	투입	로트 투입
		로트 투입 취소
		로트 공장 입고
		비생산 로트 투입
	생산	로트 준비
		로트 준비 취소
		로트 시작
		로트 시작 취소
		로트 완료
		배치(batch) 생성
		배치 준비
		배치 시작
	배치 완료	
	입고/출하	로트 입고
		로트 출하
		로트 반품
		로트 재작업
		로트 재고 변경
	속성 변경	로트 제품 변경
		로트 공정 변경
		로트 공정 스킵
		로트 타입 변경
		로트 속성 변경
	홀드/릴리즈	로트 홀드
로트 릴리즈		
수량 변경	로트 스크랩	
	로트 언스크랩	
	로트 스프릿	
	로트 머지	
	로트 수량 이동	
리웍	로트 리웍 시작	
	로트 리웍 완료	
기타	작업 예약	
	다음 장비 예약	
	장비 그룹 이동	
	측정 주기 연동(Interlock)	
자동 실행	모니터링 사이클 적용	
	예약 작업 실행	
	샘플링 룰 실행	
		Q-Time 룰 실행

확장기능

nanoFlow는 위와 같은 기능들을 컴포넌트화 하여 저장해 놓은 nanoTrack이라는 로트 추적 전용 서비스 툴을 활용한다. 즉 nanoFlow에서 nanoTrack이 제공하는 기능들을 이용하여 비즈니스 로직을 구성함으로써 아래 [그림 5]처럼 로트 추적 공정흐름도를 최종적으로 작성할 수 있다. 기존의 작성된 공정흐름도도 직관적인(intuitive) 워크플로우 모델링을 통하여 요구사항의 변화에 따른 동적인 라우팅이 가능하며, 새로운 절차, 생산요구조건에 대한 반응, 고객오더의 변화 등을 포함하는 모든 변화에 대한 수용이 가능하다. 또한 재작업, 백그라운드, 웨이퍼 분류 등과 같은 복잡한 공정도 포함시킬 수 있다.

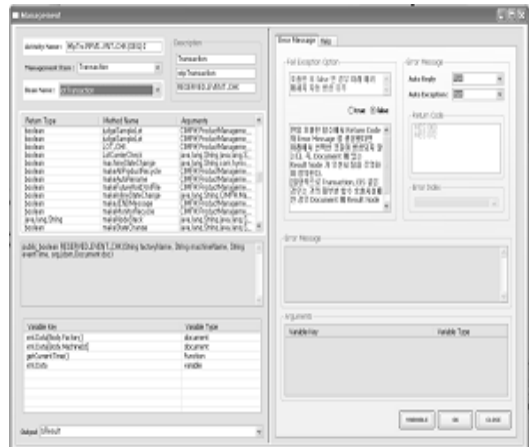


[그림 5] 로트추적 공정흐름도

nanoFlow에서는 [그림 6]처럼 기존의 제공되는 기능을 수정할 수 있으며, 특정 비즈니스 로직을 구성하는 기능을 액티비티로 구현하여 nanoTrack에 추가하여 활용할 수 있다. 예를 들어 로트 투입이라는 기능을 수정하고자 하면, 왼쪽 화면에서 make-Released(String lotName, TransitionInfo releasedInfo)라는 API를 선택하고, 그에 맞는 모델링 정보를 왼쪽 화면 하단에 입력하면 된다.

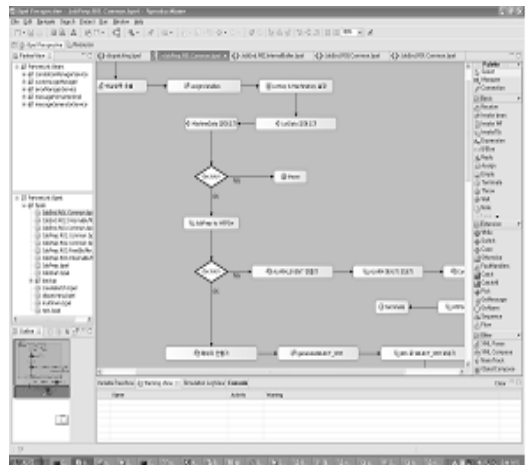
이상과 같은 프레임워크의 개방형 구조는 코드 기반의 MES 시스템 구축과 달리, 첫째, 사용자(고객 포함)가 원하는 액티비티를 설계하고 이를 쉽게 구현함으로써 맞춤형 WFMS를 도입할 수 있도록 접근한다. 둘째, 기본 및 확장 기능을 개발자

및 사용자 측면에서 바로 확인할 수 있기 때문에 개발 시간을 단축시키고 개발 오류를 사전에 차단할 수 있는 효과를 가진다.



[그림 6] 로트 추적 액티비티 관리 화면

[그림 7]은 상세 비즈니스 로직을 BPEL로 설계한 것으로 로트를 특정 공정에 진행할 수 있도록 작업 준비를 하는 프로세스를 디자인한 것이다.



[그림 7] 작업준비 상세 프로세스 설계

이것은 반도체 생산라인에서 한 로트를 특정 장비에서 작업을 하기 전에, 공정을 진행하기 위한 사전 준비 작업을 수행하는 프로세스이다. 예를

들어 다중-통과 프로세스를 제어하는 R2R(run-to-run) 컨트롤러, RMS(Recipe Management System) 시스템으로부터 진행 가능한 레시피(recipe)와 상세 레시피 파라메타에 대한 보정 기능을 수행하거나, 진행 가능한 장비 또는 장비 상태에 대한 검수 기능을 수행하는 상세 프로세스이다. 위와 같이 BPEL을 통해 상세 프로세스를 시각화를 함으로써, 쉽고 빠르게 비즈니스 로직을 개발 및 운영할 수 있다.

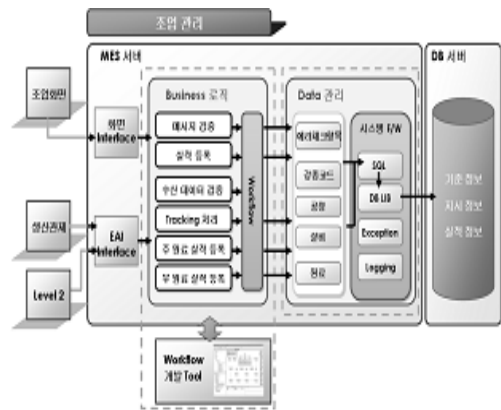
### 3.4 시스템 구현 및 효과

nanoFlow에 의하여 설계되고 테스트한 기능들은 [그림 8]처럼 “조업관리” MES 서버에 등록되어 사용되어 진다. 적용절차는 다음과 같다. 먼저 nanoFlow와 같은 IDE(Integrated Development Environment) 개발환경에서 자바 기반으로 컴포넌트와 동일한 개념인 서비스를 개발한다. 개발된 서비스는 partner link 로 등록된 후, 비즈니스 룰을 모델링(BPEL 파일 생성)한다. 서비스와 워크플로우는 시스템 및 애플리케이션을 통합하는 기준이 된다. 통합된 서비스는 실제 배포하기 전에 시뮬레이션과 디버깅을 통하여 비즈니스 룰을 사전에 검증한다. 검증이 완료되면 시스템 환경에 맞는 자동 배포 환경을 통하여 실시간으로 MES 서버에 등록된다.

MES의 비즈니스 로직은 외부 시스템(생산장비, 작업자 등)의 이벤트 단위로 구동되며, 메시지 수신, 메시지 필터링, 이벤트 관리를 통해 최종적으로 데이터베이스에 결과 데이터를 저장하게 된다. nanoFlow 기반에서는 가장 기본적인 데이터베이스 트랜잭션 관리 뿐 아니라, 위에서 언급한 비즈니스 로직을 구성하는 객체들을 틀 안에 포함시킴으로써, 전체 MES와 비즈니스 로직 구성요소를 BPEL로 구성할 수 있도록 하였다. [그림 8]에서 비즈니스 로직과 데이터 관리 부분이 여러 개의 액티비티로 구성되어져 있다.

300mm 반도체 라인에 맞는 최적화된 자동화 시스템을 구축하는 과정에서 기존의 프레임워크 대신 BPEL 기반의 WFMS 프레임워크를 MES에 적용한 결과로 다음과 같은 효과가 파악되었다. 첫

째, 비즈니스 로직 및 룰을 비 프로그래밍 방식인 워크플로우를 기반으로 개발 및 관리함으로써 개발 기간이 현격히 단축되었다. 둘째, 비주얼 모델링을 통하여 개발자 및 실무자간의 커뮤니케이션이 원활하여 지고 업무과약이 용이하게 되었다. 셋째, 유지보수 내역이 발생하더라도 수정되거나 추가될 비즈니스 로직 및 룰베이스를 따로 관리하고 있기 때문에 시스템을 정지시킬 필요가 없다. 넷째, 업무변경시 업무담당자가 직접 클라이언트 화면을 이용하여 비즈니스 로직을 변경할 수 있으며, 변경내용에 대하여 직접 테스트 및 디버깅 분석이 가능하였다. 마지막으로 표준의 채택으로 확장성 및 상호운용성이 향상되었다.



[그림 8] MES 서버 구조

새로운 프레임워크 기반의 MES 시스템이 고객사에서 운영된 지 약 1년이 지난 2009년 6월 현재, 이상에서 언급한 효과 외에 시스템의 초기 개발 기간 및 향후 유지 보수 측면을 고려해 본다면 기존 대비 30% 정도의 비용절감 효과가 있는 것으로 파악되었다. 이는 유지 보수 인력의 절감과 아울러 유지 보수 대응 속도가 현저히 빨라졌기 때문이다.

## 4. 결 론

본 연구는 MES 구축에 있어서의 WFMS 적용



에 관한 것으로, 기존의 프레임워크에서 구조적으로 존재하였던 소프트웨어의 반응성, 유연성, 통합성 및 상호운용성 등의 문제들을 새로운 환경 내에서 해결하고자 하였다. 특히 BPEL의 채택은 시스템 개발 생산성 향상은 물론 유지보수에 있어서도 많은 장점들을 보여 준다. 그러나 다음과 같은 한계점을 파악되었으며 이들은 추후 해결되어야 할 문제들이다.

첫째, 서비스를 어떻게 나누고 통합할 것인가에 대한 고민이 필요하다. 너무 작거나 크지 않은, 기능별로 중복되지 않도록 서비스에 대한 설계 가이드를 제시할 필요가 있다. 둘째, 반도체 생산라인 특성상 365일 무정지 시스템이 필요하고, 이를 위해 동적 로딩 및 서비스 관리 기술에 대한 고민이 필요하다. 이것을 해결하기 위해서는 현재 WFMS에 새로운 IT 기술을 응용할 필요가 있고, nanoFlow 차세대 버전에 OSGi(Open Service Gateway Initiatives) 기술을 접목하는 차세대 통합 프레임워크 개발을 진행 중에 있다.

## 참 고 문 헌

- [1] Balasubramanian, K., "A White Paper onFlow-briZ-A Configurable workbook framework", *Wipro Technologies*, cited, 2009.
- [2] Boucher, T. and A. Yalcin, *Design of Industrial Information Systems*, Academic Press, 2006.
- [3] Kletti, J., *Manufacturing Execution System-MES*, 2007.
- [4] Li, W. and Y. Fan, "Development of ERP based on workflow management system", *Proceedings of The 3rd International Conference of Electronic Commerce*, (2003), pp.981-984.
- [5] Neumann S., H. Hansmann, and J. Becker, "Workflow-Integrated ERP : An Architecture Model for Optimized Coordination of Intra- and Interorganizational Production Planning and Control", *ECIS*, (2002), pp.991-1004.
- [6] Stoilov, T. and K. Stoilova, "E-business workflow modeling and execution tools", *International Conference Automatica and Informatics*, 2006.
- [7] Thakur A. and V. Kolhar, "An Approach to Integrating Product Lifecycle with Manufacturing Execution and Supply Chain Processes", [www.cioindex.com/portal/cio\\_community/cio\\_whitepapers](http://www.cioindex.com/portal/cio_community/cio_whitepapers), 2009.
- [8] Workflow Management Coalition : *Terminology and Glossary*, Document Number WF MC-TC-1011, 1999.
- [9] Yu. X., S. Guo, and M. Yang, "Workflow-Based ERP Research in Automobile Refitting Enterprise", *1st International Workshop on Education Technology and Computer Science*, Vol.2(2009), pp.287-290.

## ◆ 저 자 소 개 ◆



**박 동 진 (mispdj@kongju.ac.kr)**

현재 공주대학교 산업시스템공학과 교수로 재직 중이며, 아주대학교 산업공학과를 졸업하고 한국외국어대학 및 아주대학교에서 경영정보시스템(MIS) 전공으로 석사 및 박사를 취득하였다. 박사 후에는 미국 보스턴 대학교, 위스콘신 대학교, 마켓대학교 등에서 연구를 하였으며, 주요 경력으로는 한국생산성본부 선임연구원과 남서울대학교에서 조교수로 근무하였다. 한국경영과학회지, 정보처리학회지 등 다수의 경영학 및 컴퓨터공학 관련 학회지에 논문을 게재한 바 있다. 주요 관심분야는 생산정보시스템(ERP/MES/자동화시스템), 메타데이터 및 시맨틱웹, 지식기반시스템, 데이터베이스 설계, 웹 기술 개발 등이다.



**장 병 훈 (fajbh@naver.com)**

고려대학교 응용통계학과를 졸업하였다. 하이닉스(구 현대전자)를 거쳐 현재 공장자동화 전문화 업체인 에임시스템(주) R&D 센터장으로 재직 중이다. 주로 하이테크 산업 자동화 제품 및 솔루션 개발을 담당하고 있다. 주요 개발 제품은 SEComEnabler(SECS Driver), nanoMC(Machine Controller), nanoDSP(Dispatcher), nanoTrack(MES), nanoFlow(Workflow), nanoFrame(Framework) 등이다. 주요 관심분야는 e-Manufacturing, SOA, IMS(Intelligent Manufacturing System) 등이다.