

제조시스템자동화에 있어서 BPEL 기반 워크플로우 관리시스템의 적용

박동진*, Jaejin Jang**, 장병훈***, 김수경***

*공주대학교 산업시스템공학과,

** Dept. of Industrial Engineering, U. of Wisconsin, Milwaukee,

*** 에임시스템(주)

The Implementation of BPEL based Workflow Management System in Manufacturing System Automation

Dongjin Park, Jaejin Jang, Byoung-Hoon Jang, Soo Kyoung Kim

Kongju National Univ., Univ. of Wisconsin, AIM Systems, Inc.

E-mail : mispdj@kongju.ac.kr

Abstract

This paper outlines opportunities and challenges in the Implementation of BPEL based WFMS(Work Flow Management System) for the MES(Manufacturing Execution Systems) level in semiconductor manufacturing. At present, the most MES that are composed of several hundreds of applications in semiconductor wafer fabrication shop have the same problems as others about flexibility 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 inflexible communication infrastructure among the manufacturing process components and the difficulty in porting existing application software to new configurations. In this paper, the issues about BPEL standard, used for the flexibility of Workflow Management System, are presented. We introduce the integrated development framework named nanoFlow which is optimized for developing the BPEL based WFMS application for automated manufacturing system. We describe a WFMS implemented with using nanoFlow framework, review and evaluate the system in terms of flexibility and adaptability.

1. 서론

현대의 제조시스템은 잦은 요구사항의 변화, 리드타임의 압력, 새로운 기기의 도입에 따른 호환성 등 수많은 문제에 직면하고 있다. 기업이 제조경쟁

력을 확보하기 위해서는 제조의 동기화가 매우 중요하다. 이를 위해서 제조실행시스템(MES: Manufacturing Execution System)에 많은 투자를 하고 있다. MES은 특히 소프트웨어 측면에서 제조환경의 변화에 대하여 유연하게 대처할 수 있어야 하며, 이질적인 컴포넌트들을 효과적으로 통합

할 수 있는 능력을 확보하는 것이 매우 중요하다.

반도체 기업의 웨이퍼 제조시스템도 마찬가지다. Shop floor에서 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 장애요인이 되기도 한다. 특히 현장의 소프트웨어 구성은 매우 이질적이어서 많은 응용프로그램이 서로 다른 운영체제하에서, 다양한 데이터베이스를 이용하고, 다양한 프로토콜을 통하여 커뮤니케이션을 하고 있다. 이것이 결국 엄청난 통합 및 유지보수 비용을 초래하기도 한다.

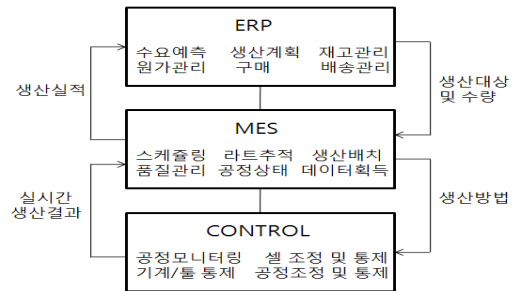
본 연구는 워크플로우(workflow) 기술을 적용하여, 이상에서 언급된 MES 수준에서의 소프트웨어 이슈들을 해결하는 방안을 제시한다. 구체적으로 제2장에서는 MES수준에서 워크플로우 관리시스템(WFMS: Workflow Management System) 도입의 필요성을 주장하였으며, 워크플로우 표준 언어로서 BPEL을 채택한 WFMS의 아키텍처를 기술하였다. 제3장에서는 반도체 제조시스템의 WFMS 솔루션인 nanoFlow를 소개하고 MES 주요 기능중의 하나인 Lot Tracking을 구현하는 과정을 소개한다. 또한 실제 구현한 시스템을 통한 경험과 효과를 기술하고, 마지막으로 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(Boucher and Yalcin, 2006). 각 수준에는 다양한 기능들이 포함되어 있으며, 이들이 정보시스템으로 인하여 수평으로 그리고 수직적으로 서로 통합됨으로 인해서 제조생산활동을 위한 정보를 원활하게 제공하는 것이다. <그림 2-1>은 각 수준들의 관계와 상세기능들을 보여준다. 여기서 MES는 ERP로부터 몇 주 단위의 생산

계획을 전달받으며, 생산이 진행되면서 생산 실적을 ERP로 피드백 한다. 또한 MES는 제조현장에서의 자세한 생산정보를 다루며, 하위의 Control 수준에 있는 기계 컨트롤러에 특정 부품을 어떻게 처리할 것인가에 대한 자세한 정보를 내려 보내기도 한다.



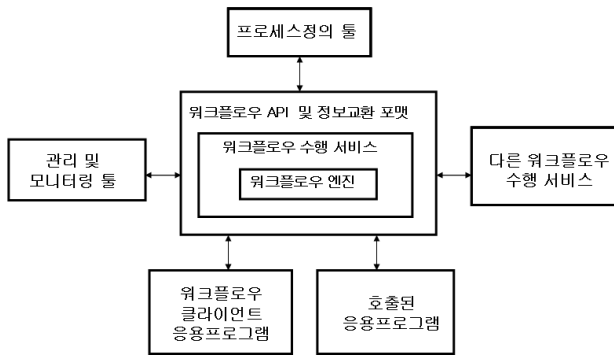
<그림 2-1> 제조기업의 정보체계수준 및 기능

최근 MES 솔루션 개발자와 컨설턴트들은 MES가 미래의 변화에 대응하고 유연하게 적응하기 위해서는 다음과 같은 요구사항을 만족시키는 소프트웨어 아키텍처를 가져야 한다고 주장한다(Kletti, 2007). 첫째, ERP 하단에 있는 모든 기능들을 수평적으로 통합할 수 있는 기반을 갖추어야 한다. 둘째, 표준을 채택하며 요구사항에 따라 확장가능한 구조여야 한다. 셋째, 프로세스 및 상세 요구사항에 표준 모듈을 쉽게 적용가능한 구조여야 한다. 넷째, 모든 수준에 걸쳐 표준화된 인터페이스의 채택하여야 한다.

WFMS를 포함한 워크플로우 기술은 주로 ERP 수준의 비즈니스 프로세스의 모델링, 리엔지니어링 그리고 자동화에 초점이 맞추어져 왔다. 그러나 WFMS의 다음과 같은 장점들로 인해서 MES에서 충분히 적용되어 질 수 있음을 알 수 있다. 첫째, 응용프로그램으로 부터 비즈니스로직 및 비즈니스 룰을 분리함으로써 업무가 변경되더라도 프로그램의 수정이 용이하다. 둘째, 비슈얼한 프로세스 모델링 툴의 이용으로 최종사용자가 직접 비즈니스 로직을 관리할 수 있다. 셋째, 표준을 기반으로한 시스템의 확장이 가능하다. 넷째, 각 컴포넌트들을 독립적으로 개발하고 재사용할 수 있다. 이러한 특성으로 인해서 최근 들어 제조기업의 MES 수준에도 WFMS가 적용되기 시작하고 있다(Balasubramanian, 2009).

2.2 반도체 제조시스템에서 WFMS의 적용

워크플로우 참조 모델(WfMC, 2009)은 WFMS의 기능적 구성요소들과 이들 간의 인터페이스를 포함한 WFMS의 본원적 구조를 보여준다. <그림 2-2>에서처럼 구성요소들은 프로세스 정의 툴, 워크플로우 엔진을 포함한 워크로우 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(WfMC, 1999).



<그림 2-2> WfMC 참조 모형

반도체 제조시스템에 있어서 제조프로세스는 매우 복잡하다. 웨이퍼를 생산하기 위해서는 수백가지의 세부단계를 거친다. 생산현장의 관리 프로세스들을 살펴보면 생산운영계획, 자원할당 및 스케줄링, 라트 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 룰이 적용되기도 한다. 이상의 모든 프로세스에 WFMS가 적용됨으로써 반도체 제조시스템에서 소프트웨어의 통합(integration), 반응성(responsibility)과 적응성(adaptability) 등을 확보할 수 있다. 아래 <표 2-1>은 반도체 제조시스템 WFMS에 특화된 요구사항을 정리한 것이다.

<표 2-1> 반도체 분야의 WFMS 요구사항

분야	요구사항
기본 기능	병렬처리 (비)동기처리 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 Exception 관리
커뮤니케이션 패턴	(비)동기처리 추상화된 미들웨어 기능 전, 후 메시지 연관 기능 Legacy 시스템 Integration 기능 메시지 Dispatching 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 엔티티가 없으면서 비즈니스 엔티티들 간에 장기적이고 계속적인 상태(long-lived state)에서 서로 협조하면서 이벤트가 발생할 시 각자가 자기역할을 함으로써 프로세스를 처리하는 기술이다. 즉 peer-to-peer 방식으로 협업하는 모델이다. 반면에 Orchestration 은 하나의 비즈니스 엔티티에 초점을 맞추는 것으로, 특정지점 한곳에서 사전에 정의된 대로 프로세스의 전체를 통제하는 것이다. 즉 비즈니스를 주도하는 하나의 참여자 관점에서 여러 프로세스들을 통합하는 모델이다(Stoilov and Stoilova, 2006).

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

프로세스간에 통합이 가능한 표준 인터페이스가 필요하다. 최근 WFMS 솔루션들은 내부적으로 프로세스를 표현하는 XML 기반의 프로세스정의 언어를 채택하는데, 그 중에서 BPEL이 Orchestration 형식의 서비스를 지원할 뿐 아니라 표준인터페이스를 제공함으로써 프로세스의 수평적 그리고 수직적 통합을 가능케 한다.

BPEL은 웹서비스를 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 2-3>는 BPEL 프로세스 정의의 간단한 예이다.

```
<process name="BC01workFlow" ... properties..>
  <partnerLinks>> //파트너 정의(WSDL 파일로부터
    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 2- 3> BPEL 프로세스의 예

BPEL 프로세스는 기본적으로 실행단계인 <invoke>와 <assign>과 같은 액티비티(activity)와 <sequence>와 같이 액티비티를 포함하는 컨테이너로 구성된다. 또한 <그림 2-3>는 액티비티간에 메시지를 어떻게 교환하는 가를 보여준다. 메시지는 기본적으로 복잡한 변수로서 표현되며, <receive>, <invoke>, <assign>, 그리고 <reply> 액티비티들은 메시지와 관련된 문장들이다.

3. 반도체기업 MES에 WFMS의 적용

3.1 워크플로우 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 오류발생 가능성이 높다. 둘째, 모듈간 인터페이스를 위한 메세지 전달 영역과 비즈니스 영역이 혼재되어 유지보수 및 관리가 어렵다. 셋째, 인터페이스를 위한 표준의 채택이 없어서 모듈의 재사용성은 물론 통합성 및 이식성이 부족하다.

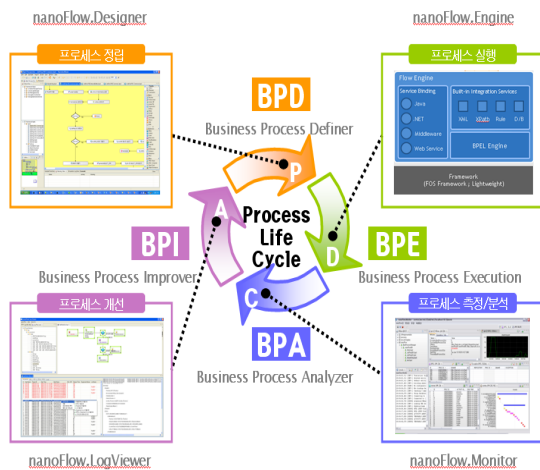
이상과 같은 문제를 해결하는 방법은 새로운 개발 프레임워크의 도입이 필요하다. 즉 워크플로우 형태의 가시성을 확보하여 생산운영 로직의 분석을 용이하게 하고 개발생산성을 높이고 신속한 변경요구 대응과 오류발생 가능성을 최소화하는 프레임워크의 도입이 필요하다. 또한 프레임워크는 응용프로그램 통합을 위한 표준인터페이스를 제공하며 프로그램의 관리 및 모니터링이 가능하여야 한다.

3.2 nanoFlow 아키텍처

nanoFlow는 에임시스템(주)에서 개발한 범용 워크플로우 솔루션으로 공장용 응용시스템 개발에 최적화되어 있다. Java 기반으로 구현되었으며 프로세스 표준언어로 BPEL을 채택한다. 솔루션은 <그림 3-1>과 같이 크게 개발환경(nanoFlow.Deginer), 운용환경(nanoFlow.Engine), 관리환경(nanoFlow..LogViewer), 그리고 모니터링 환경(nanoFlow.Monitor)으로 구성된다. 각 구성 요소들은 약결합(loosely coupled)되어 있어서 독자

적인 운영이 가능하고 메시지 전달이나 원격프로시저 호출 등에 의해서 상호연결 및 결합이 가능하다.

nanoFlow.Deginer는 Drag-Drop방식으로 비즈니스로직을 비주얼모델링 할 수 있으며, 완성된 모델을 실시간으로 테스트가 가능하다. nanoFlow.Engine은 응용프로그램을 통합하고 비즈니스 로직을 관리하는 실행 주체로서 서비스 바인딩(service binding), 통합서비스기능(Built-in Integration Service), BPEL 엔진 등으로 구성된다. nanoFlow.Monitor는 nanoFlow 기반의 모든 응용프로그램을 실시간으로 모니터링하고, 비즈니스의 로직을 추적하고, 결과와 통계치를 생성한다. nanoFlow.LogViewer는 애플리케이션이 생성하는 로그 파일을 기준으로 BPEL의 성공여부, Elapsed Time 에 대한 최대, 최소, 평균값을 다양한 형태로 보여줌으로써, 향후 발생될 문제를 사전에 예측하고 방지할 수 있다.



<그림 3-1> nanoFlow 아키텍처

3.3 Lot Tracking 프로세스의 설계

Fab의 생산은 매우 복잡한 공정을 거치며 각 제품품은 Lot단위로 관리되어진다. MES의 Lot Tracking 기능은 Lot를 생성(Lot creation)하며, 각 세부공정상에 있는 Lot의 공정이력정보를 포함하며, 생산 공정상에 있는 Lot들의 가시성(visibility)과 통제를 가능하게 한다. 구체적으로 <표 3-1>과

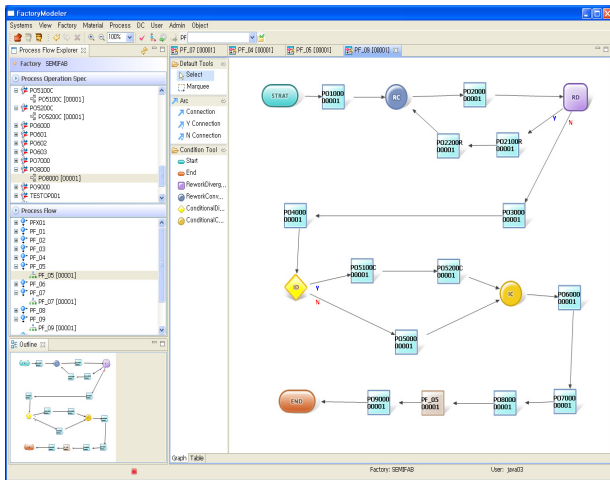
같은 기능들이 포함한다.

<표 3-1> Lot Tracking 상세 기능

생산 지시	WO	WO 시작 WO 완료				
기본기능	투입	Lot 투입 Lot 투입 취소 Lot 공장 입고 비생산 Lot 투입				
		생산	Lot 준비 Lot 준비 취소 Lot 시작 Lot 시작 취소 Lot 완료 Batch 생성 Batch 준비 Batch 시작 Batch 완료			
			입고/출하	Lot 입고 Lot 출하 Lot 반품 Lot 재작업 Lot 재고 변경		
				확장기능	속성 변경	Lot 제품 변경 Lot 공정 변경 Lot 공정 스킵 Lot 타입 변경 Lot 속성 변경 Lot ID 변경
						홀드/릴리리즈
	수량 변경					
						리워
			기타			
					자동 실행	Future Action 실행 Sampling Rule 실행 Q-Time Rule 실행

위와 같은 기능은 예임시스템(주)에서 개발한 MES 제품인 nanoTrack에서 제공하고 있으며, nanoFlow 는 nanoTrack 이 제공하는 기능과 실제 고객별, 산업별로 다양한 형태의 비즈니스 로직을 쉽게 구현하고 관리할 수 있도록 최적화되어 있다. 이것은 일반적인 IT 벤더들이 접근하는

WFMS 방식과 차별화될 수 있는 부분이기도 하다.



<그림 3-2> 공정흐름도

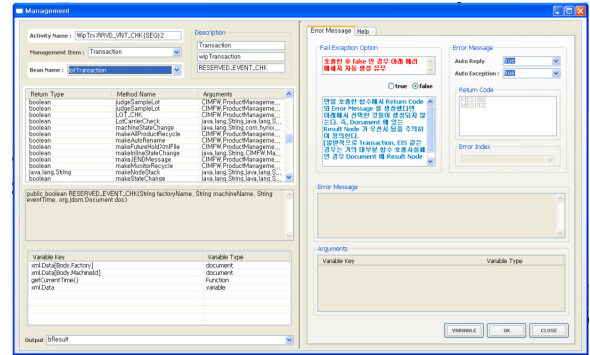
nanoTrack에서는 <그림 3-2>처럼 Lot에 적용되는 전체 공정흐름도를 작성할 수 있다.

Intuitive 워크플로우 모델링을 통하여 요구사항의 변화에 따른 동적인 라우팅이 가능하며, 새로운 절차, 생산요구조건에 대한 반응, 고객오더의 변화등을 포함하는 모든 변화에 대한 수용이 가능하다. 또한 재작업, 백 그라인드, 웨이퍼 분류 등과 같은 복잡한 공정도 포함시킬 수 있다.

nanoFlow에서는 <그림 3-3> 처럼 nanoTrack에서 제공하는 기능뿐 만아니라, 특정 비즈니스 로직을 구성하는 기능을 추가 액티비티로 구현함으로써, 코드 기반의 MES 시스템 구축의 단점을 보완할 수 있다.

이것은 Lot Tracking을 하기 위한 기본, 확장 기능을 개발자 및 사용자 측면에서 바로 확인할 수 있기 때문에 개발 시간을 단축시키고, 개발 오류를 사전에 차단할 수 있는 효과를 가진다.

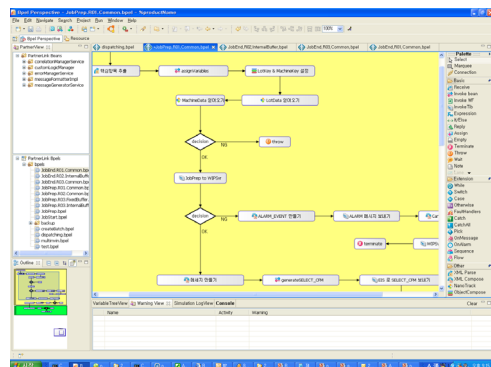
nanoFlow는 개방형 구조로써, 사용자(고객 포함)가 원하는 액티비티를 설계하고 이를 쉽게 구현함으로써, 맞춤형 WFMS을 도입할 수 있도록 접근한다. 이것은 앞서 설명한 바와 같이 대형 IT 벤더들이 접근하는 방식과 차별화되는 점이기도 하다.



<그림 3-3> Lot Tracking 액티비티 화면

라인을 운영하기 위한 기본, 확장 기능 항목들이 왼쪽 화면에 표시가 되고, 사용자는 해당 기능을 선택 후, 요구되는 모델링 정보를 입력한다. 예를 들어, Lot 투입이라면, 왼쪽 화면에 makeReleased (String lotName, TransitionInfo releasedInfo) 라는 API를 선택하고, 그에 맞는 모델링 정보를 왼쪽 화면 하단에 입력한다.

<그림 3-4>는 Lot tracking을 위한 상세 비즈니스 로직을 BPEL로 설계한 것으로 Lot을 특정 공정에 진행할 수 있는 지 작업 준비를 하는 프로세스를 디자인한 것이다.

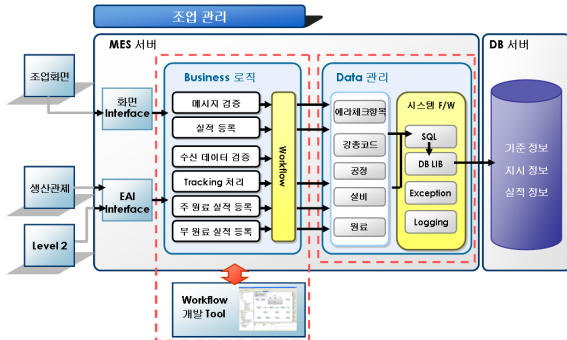


<그림 3-4> 작업준비 프로세스 설계

반도체 생산라인에서 Lot을 특정 장비에서 작업을 하기 전, 공정을 진행하기 위한 사전 준비작업을 수행하게 된다. 예를 들어 R2R, RMS 시스템으로부터 진행 가능한 Recipe와 상세 Recipe Parameter에 대한 보정 기능을 수행하거나, 진행 가능한 장비 또는 장비 상태에 대한 Validation 기능을 수행한다. 위와 같이 BPEL을 통해 시각화를 함으로써, 쉽고 빠르게 비즈니스 로직을 개발 및 운영할 수 있다.

3.4 시스템 구현 및 효과

nanoFlow에 의하여 설계되고 테스트한 기능들은 <그림 3-5>처럼 “조업관리” MES 서버에 등록되어 사용되어 진다. 적용절차는 다음과 같다. 먼저 nanoFlow (Eclipse Plug-In) 와 같은 IDE 개발 환경에서 Java 기반으로 컴포넌트와 동일한 개념인 서비스를 개발한다. 다음으로 개발된 서비스는 partner link 로 등록되어, 비즈니스 로직을 그림 그리듯이 개발한다. 개발된 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 자동배포 환경을 통하여 실시간으로 서비스를 MES 서버에 등록한다.



<그림 3-5> MES 서버 구조

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

4. 결론

본 연구는 MES 구축에 있어서의 WFMS 적용에 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 문제들 이다.

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.

[참고문헌]

- [1] Balasubramanian K. A White Paper on Flow-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, Springer, 2007.
- [4] Stoilov, T and K. Stoilova, E-business workflow modelling and execution tools, International Conference "Automatica & Informatics", October 3-6, 2006, Sofia, Bulgaria
- [5] Workflow Management Coalition, Workflow Reference Model, Homepage of Workflow Management Coalition, 2009.
- [6] Workflow Management Coalition: Terminology & Glossary, Document Number WFMC-TC-1011, Feb. 1999