

SEMI E40-0312

프로세싱 관리 표준 (STANDARD FOR PROCESSING MANAGEMENT)

이 표준은 기술적으로 Global Information & Control Technical Committee 에 의해 승인되었으며, 해당 버전의 출판은 2011 년 12 월 24 일 Global Audits and Reviews Subcommittee 에 의해 승인되었다. 1995 년에 초본이 출판되었고, 이전 버전은 2011 년 11 월에 출판되었다. 이번 출판본은 www.semiviews.org 와 www.semi.org 에서 2012 년 3 월부터 이용 가능하다.

NOTICE: This translation is a REFERENCE COPY ONLY. If differences should exist between the English version and a translation in any other language, the English version is the official and authoritative version.

본 번역서는 참고용으로써, 영문원본과 한글번역본간에 차이가 발생할 경우, 영문원본을 우선으로 해야함을 알려드립니다. 본 번역서에 대해 문의사항이 있는 경우 krstandards@semi.org 로 문의바랍니다.

1 목적 (Purpose)

- 1.1 장비에 존재하는 재료 프로세싱의 관리와 명령(command)을 자동화하는것은 공장 자동화를 가능하게 하는 중요한 측면으로, 이 표준에서는 장비내의 재료 프로세싱에 관한 반도체 제조 환경에 필요한 통신을 다룬다.
- 1.2 이 표준은 프로세싱 에이전트가 수신한 지정된 재료에 대해 적절한 프로세싱의 응용 프로그램을 명시하고, 재료 프로세싱(material processing)의 개념, 프로세싱과 관련된 장비의 동작 및 작업(task)을 수행하는데 필요한 메시징 서비스(messaging services) 들을 설명한다.
- 1.3 여기서 정의하는 통신 서비스는 독립적인 시스템의 표준에 근거하는 상호 운용을 가능하게 하고, 이를 통해 개발되는 응용프로그램이 이러한 서비스의 존재를 추측할수 있게하고 소프트웨어 제품들이 통신서비스를 제공할 수 있도록 개발되게 한다. .
- 1.4 자동 프로세싱 관리의 실행으로 재료에 대한 오처리(misprocessing)를 없애고, 명문화된 표준의 적용으로 장비 구성요소의 통합에 필요한 노력을 크게 줄여 프로세싱 설정 시간을 감소시킨다. 준수 해야할 항목은 최소한으로 줄이지만 그 기준은 매우 명확해야만 한다.

2 범위 (Scope)

- 2.1 이 표준의 범위는 개별 프로세스 잡(processing jobs)에 기반한 자동화된 재료 프로세싱이며, 이는 클러스터 도구 내의 모듈에 대한 프로세싱 관리에 필요한 기능을 제공한다. 다양한 유형의 장비에 대한 호스트 컨트롤(control)뿐만 아니라, 그외 다른 다중-리소스(multi-resource) 설비의 하위 시스템에도 적용된다.

2.2 이 표준은 그룹 내에서 재료의 동일한 처리와 독립적인 그룹들의 병렬 프로세싱에 대한 개별적인 작업(jobs) 관리를 지원한다. 재료가 다른 재료를 담는 경우 (웨이퍼를 담는 캐리어와 같은), 각각의 재료 유형에 따라 프로세싱이 명시되어야만 한다.

2.3 단순한 튜닝(tuning) 메카니즘은 프로세스 단계 간에 제한적인 피드포워드(feedforward)와 피드백(feedback)을 위해 제공되고, 메소드(Method)가 레시피 가변 매개변수(recipe variable parameters)들을 이용하기 위해 정의되지만, 이들이 모든 폐쇄된 순환 컨트롤(closed loop control) 요구를 만족시킬것 이라고 기대되지는 않는다. 복잡한 데이터의 처리(handling)와 튜닝(tuning) 지원에 대해서는 그외 다른 메커니즘들이 보다 큰 유연성을 갖을 것이라 기대된다.

2.4 이 표준은 프로세싱을 위해 재료를 수신하거나, 프로세싱이 완료된 후 재료의 처리(disposing)와 같은 서비스를 제공하지 않는다. 재료 이동 자동화는 다른 적용가능한 SEMI 표준에 정의된 내용들과 같이 본 표준과는 다른 서비스를 통해 제공될 수 있다.

2.5 이 표준에서는 장비와 관련된 개념과 동작부터 메시징 서비스를 아우르는 솔루션을 제시하지만 메시징 프로토콜을 정의하지는 않는다.

2.6 메시징 서비스는 메시지 교환에 대한 식별과 해당 메시지에 포함된 데이터 정의를 포함하지만, 메시지의 구조에 대한 정보, 메시지내의 데이터 표현 방식, 혹은 메시지 교환 방식을 포함하지는 않는다. 추가적인 정보는 메시징 프로토콜에 포함된다.

2.7 정의된 서비스들은 다중 프로토콜에 적용되며, 특별한 프로토콜로 (예: SECS-II) 프로세싱 관리 서비스를 연결하는 정보는 추가적인 부록 표준에서 다루지게 된다.

2.8 서비스는 서비스 사용자와 서비스 공급자 사이에 신뢰할 만한 연결이 설정되는 통신 환경을 말한다. 서비스는, 설정, 유지, 연결 해제 및, 통신 실패 처리는 이 표준에서 다루지 않는다.

NOTICE: SEMI Standards and Safety Guidelines do not purport to address all safety issues associated with their use. It is the responsibility of the users of the Documents to establish appropriate safety and health practices, and determine the applicability of regulatory or other limitations prior to use.

이 표준은 사용에 관련된 안전 사항의 명시를 의미하지 않는다. 사용자는 이 표준을 사용하기에 앞서 안전과 건강에 관련된 규제 방안을 수립하고, 규제나 기타 제한사항들에 대한 적용 방안을 규정할 책임이 있다.

3 참조 문헌 및 표준(Referenced Standards and Documents)

3.1 SEMI 표준들과 안전 지침서(SEMI Standards and Safety Guidelines)

SEMI E5 — SEMI Equipment Communications Standard 2 Message Content (SECS-II)

SEMI E30 — Generic Model for Communications and Control of Manufacturing Equipment (GEM)

SEMI E39 — Object Services Standard: Concepts, Behavior, and Services

NOTICE: Unless otherwise indicated, all documents cited shall be the latest published versions.

별도의 표시가 없으면, 인용된 모든 서류는 가장 최근에 발행된 버전이다.

4 용어 (Terminology)

4.1 다음의 정의는 알파벳 순서로 정렬되었고, 일부 정의는 이 절 이외 다른 곳에서 정의된 용어를 사용한다. 이들 용어의 기본적인 이해를 위해 이 절 이외의 참조는 필요하지 않다.

4.2 정의 (Definitions)

4.2.1 에이전트 (*agent*) — 하나 혹은 그 이상의 서비스 자원(resource)을 제공하고 다른 에이전트들의 서비스를 사용하는 공장 내 지능형 시스템. 호스트, 장비, 셀(cell), 클러스터(cluster), 클러스터 모듈(cluster module), 스테이션 컨트롤러(station controller), 워크스테이션(workstation)를 일반화한 표현. 에이전트들은 컴퓨터 플랫폼을 포함하며, 물리적인 시스템과 물리적인 시스템의 집합을 포함하기도 한다.

4.2.2 형식 (*form*) — 객체 속성 또는 서비스 메시지 매개변수(parameter)에 포함되는 정보를 나타내는 데이터의 유형.

4.2.3 필수 요구사항 (*fundamental requirements*) — 표준의 준수를 위해 충족되어야만 하는 정보와 동작에 대한 요구 사항, 기본적인 요구 사항은 응용 프로그램, 객체 혹은 서비스의 특정한 영역에 적용된다.

4.2.4 후 처리 (*post-conditioning*) — 해당 재료 프로세싱과 관련되지만 프로세싱이 완료된 재료가 떠난 후 프로세싱 자원(resource)에 의해 수행되는 활동이다 (예: cleanup).

4.2.5 예비 조건 (*pre-conditioning*) — 해당 재료 프로세싱과 관련되지만 프로세싱될 재료의 도착 전 프로세싱 자원(resource)에 의해 수행되는 활동이다.

4.2.6 프로세싱 에이전트 (*processing agent*) — 재료에 제조를 통한 부가가치를 독립적으로 제공할 수 있는 공장 내 지능형 시스템을 말한다.

4.2.7 프로세싱 자원 (*processing resource*) — 재료에 제조를 통한 부가가치를 제공하는 프로세싱 에이전트 내 개체를 말한다.

4.2.8 프로세스 잡 (*process job*) — 재료에 적용될 프로세싱을 명시하고 트래킹(tracking)하는 프로세싱 자원에 대한 재료의 프로세싱 잡(processing job).

4.2.9 레시피 (*recipe*) — 재료가 노출되는 프로세싱 환경을 결정하는 프로세싱 자원(resource)의 컨트롤 하에 지시(instructions), 설정 및, 매개변수 (parameters) 집합중 사전 계획되고 재사용되는 부분을 말하며, 운영과 프로세싱 주기 사이에 변경될 수 있다.

4.2.10 레시피 실행자 (*recipe executor*) — 레시피를 저장하고 진행하는 모듈의 구성요소이다.

4.2.11 레시피 네임스페이스 (*recipe namespace*) — 레시피의 저장과 관리에 대한 책임을 가지는 논리적인 관리 도메인이며, 레시피 식별자에 대한 고유함을 보장하고 그 범위 내에 저장되는 레시피에 관련된 서비스를 제공한다.

4.2.12 서비스 (*service*) — 특별한 기능에 원격 접근을 가능케 하는 서비스 제공자의 동작에 대한 메시지와 정의의 집합.

4.2.13 서비스 제공자 (*service-provider*) — 원격으로 접근가능한 특별한 기능의 제공자인 소프트웨어 컨트롤(control) 개체.

4.2.14 서비스 사용자 (*service-user*) — 관련된 서비스의 사용자인 소프트웨어 컨트롤(control) 개체.

4.2.15 관리자 (*supervisor*) — 하나 혹은 다수의 프로세싱 자원(resource)에 대해 감시적인 컨트롤(control) 책임을 가지는 개체 혹은 개체들이며, 프로세싱 관리 서비스의 서비스- 사용자이다.

4.2.16 튜닝 (*tuning*) — 특정 프로세싱 목표를 달성하기 위해 사용된 사전 정의된 레시피를 보완하는 정보의 규격.

4.3 데이터 유형 (*Data Type*)

4.3.1 불대수 (*Boolean*) — TRUE 혹은 FALSE 에 상응하는, 두개의 가능한 값 중 하나를 가진다.

4.3.2 계수 (*enumerated*) — 가능한 값의 제한된 집합 중 하나의 값을 가지며, 이들 값은 논리적 이름이 주어질 수 있지만, 단일 아이템 데이터 유형으로 표현될 수 있다.

4.3.3 형태 (*form*) — 데이터의 유형: 양수, 무부호 정수, 정수, 계수, 불대수, 텍스트, 형식화된 텍스트, 구조, 리스트, 순서화된 리스트.

4.3.4 형식화된 텍스트 (*formatted text*) — 제한된 형식을 가진 텍스트 문자열. 이는 위치(position)나 특수 문자의 사용 혹은 둘 모두를 제한 할 수 있다.

4.3.5 정수 (*integer*) — 음수 혹은 무부호 정수의 값을 가지며, 메시징 프로토콜은 가능한 값의 범위를 제한할 수 있다.

4.3.6 리스트 (*list*) — 동일한 형태를 가지는 하나 혹은 그 이상의 아이템 집합(위에 형태 중에 하나).

4.3.7 순서화된 리스트(*ordered list*) — 중요한 순서로 정렬된 리스트.

4.3.8 양의 정수 (*positive integer*) — 전체 숫자중 양의 값을 가진 모든 수, 메시징 프로토콜은 가능한 값의 범위를 제한 할 수 있다.

4.3.9 구조 (*structure*) — 특정한 배열 내에서 특정한 아이템의 집합, 혹은 혼합된 데이터 유형으로, 구성된 복잡한 구조.

4.3.10 텍스트 (*text*) — 텍스트 문자열. 메시지 프로토콜은 길이 혹은 ASCII 표현등과 같은 제한을 두기도 한다.

4.3.11 무부호 정수 (*unsigned integer*) — 양의 값을 가지는 정수나 정수 혹은 0, 메시지 프로토콜은 가능한 값의 범위를 제한한다.

5 규약 (Conventions)

5.1 *Harel State Model* — 이 문서는 정의된 객체의 다이나믹한 운용을 기술하는 Harel State Chart 표기법을 사용하며, 그에 대한 내용은 SEMI30 별첨에서 확인할 수 있다. 표기법의 공식적인 정의는 *Science of*

Computer Programming 8, “Statecharts: A Visual Formalism for Complex Systems” by D. Harel, 1987 에 나타나있다.

5.1.1 Harel 표기법은 전이중인 개체를 나타내는 상태모델의 “생성(creation)”과 “삭제(deletion)”의 개념을 포함하지 않는다. 이 문서에 기술된 “작업(job)”은 이러한 개체이며, 여기서 생성된 각각의 새로운 작업(job)은 동일한 상태모델을 복제(copy)하여 사용한다. 이 문서에서, 타원형(oval)은 개체의 생성과 그 개체의 삭제를 보여주는데 사용된다.

5.1.2 전환 테이블은 각 상태 전환의 본질을 분명하게 설명하는 상태 다이어그램과 함께 제공된다. 전환은 전환(Transition) #의 열, 현재 상태(Current State), 트리거(trigger), 새로운 상태(New State), 동작(Action(s))을 포함한다. ‘current’ 상태(state) 에서 전환에 대한 ‘트리거(trigger)’(열 3)가 발생한다. ‘actions’(열 5)는 (1) 현재 상태(current state)가 종료 됨에 따라 취해지는 동작, (2) 새로운 상태(new state)로 진입하게 됨에 따라 취해지는 동작, (3) 전환과 가장 밀접하게 관련되어 일어나는 동작들과 같은 조합을 포함한다. 차이는 발생하지 않는다.

5.1.3 이 표준에 포함된 상태모델은 프로세싱 관리(Processing Management) 준수를 위한 필수 사항이며, 상태모델은 상태 모델 다이어그램, 상태 정의, 상태 전환 테이블로 구성된다. SEMI E30 혹은 유사한 방식의 수집 이벤트를 사용할 때, 별도로 특별히 명시되어있지 않다면이 표준에서의 모든 상태 전환은 수집 이벤트에 해당한다.

5.1.4 상태모델은 장비에 대한 호스트의 관점을 나타내며 내부 장비 운용을 설명할 필요는 없다. 수집 이벤트를 사용할 때, 모든 프로세싱 관리(Processing Management) 상태모델 전환은 이들 전환의 필요조건을 충족하는 적절한 내부 장비의 수집 이벤트들과 순서에 따라 연결된다. 어떤 실행이 일어난다는 것은, 장비는 어떤 상태(state)로 진입하게 된다는 것이고, 상태(state)로의 전환을 위해 프로세싱 관리(Processing Management) 상태모델에서 요구하는 모든 조건을 이미 만족시켰다는 뜻이다. 이 경우 장비는 추가적인 동작을 하지 않고 요구된 전환을 수행한다.

5.2 객체 속성 표현 (Object Attribute Representation) — 표준화된 객체에 대한 객체 정보 모델은 아래와 같은 제목을 가지는 속성 정의 테이블로 설명된다:

Attribute Name	Definition	Access	Requirement	Form
속성 (Attributes)의 공식적인 텍스트 명칭.	설명	RO or RW	Y or N	(아래의 설명 참조)

5.2.1 접근(access) 열은 서비스-사용자의 속성에 대한 접근(access)의 성격을 나타내기 위한 것으로 RO(읽기만) 혹은 RW(읽고 쓰기)가 사용된다.

5.2.2 이 속성이 서비스에 대한 필수적인 준수 사항을 만족시키기 위해서 지원되어야하는지를 표현하기 위해, 요청 (Rqmt) 열에서 ‘Y’ 나 ‘N’으로 표시된다.

5.2.3 Form 열은 속성의 형식을 나타내기 위해 사용된다. (¶ 4.1 정의 참조).

5.3 서비스 메시지 표시 (Service Message Representation)

5.3.1 서비스 자원 정의 (Service Resource Definition) — 다음의 서비스 자원(resource) 정의 테이블은 주어진 서비스 그룹에 대한 특정 메시지를 정의한다:

<i>Message Service Name</i>	<i>Type</i>	<i>Description</i>
메시지 명칭	N or R	서비스 의미

5.3.1.1 유형(Type)은 N = 알림(Notification) 또는 R = 요청(Request) 일 수 있다.

5.3.1.2 알림 유형의 메시지는 서비스 제공자 의해 시작되고, 제공자는 사용자/요청자로부터 응답을 기대하지는 않는다.

5.3.1.3 요청 메시지는 서비스 사용자 혹은 요청자에 의해 시작된다. 요청 메시지는 데이터 혹은 제공자에게 활동을 요청하며, 특정한 응답 메시지가 있을것을 기대한다 (메시지 내용에 대한 추정 없음).

5.3.2 서비스 매개변수 디렉토리 (Service Parameter Dictionary) — 다음의 서비스 매개변수(parameter) 디렉토리 테이블은 서비스에 사용되는 매개변수(parameters)를 정의한다:

<i>Parameter</i>	<i>Form</i>	<i>Description</i>
매개변수(parameter) X	데이터 유형	매개변수(parameter) X 는 A 에서의 B 이다.

5.3.2.1 서비스의 각 매개변수(parameter)는 테이블의 행에 나타난다. 첫 번째 열은 매개변수(parameter)의 이름을 포함하며, 해당 기본 요소(primary)의 형식과 내용을 설명하는 열이 뒤따른다.

5.3.2.2 형식(Form) 열은 매개변수(parameter)에 포함된 데이터 유형을 나타낼 때 사용된다 (§4.2 정의 참조).

5.3.2.3 서비스 매개변수 디렉토리 테이블(Service Parameter Dictionary)의 설명(Description) 열은 매개변수(parameter)의 의미, 그것이 가지는 값, 다른 매개변수(parameters)들과의 상호 관계를 설명한다.

5.3.2.4 수 많은 매개변수(parameter)들이 'XxxList' 로 정의되는 것을 방지하기 위해, 이 문서는 '(List of) Xxx' 로 리스트를 명명하는 규약을 도입한다. 이 경우 리스트가 아닌 변수 Xxx 로 정의된다. 'list' 는 동일한 데이터 유형의 0 또는 그 이상의 아이템의 집합(collection)을 나타낸다. 여기서 리스트가 요청과 응답을 모두 사용하는 경우, 리스트 상의 요청순서는 응답에서도 유지된다. 리스트내에 항목이 하나도 존재하지 않아도 된다고 특별히 허용하지 않는 한, 적어도 하나의 항목은 포함되어야 한다.

5.3.3 서비스 메시지 정의 (Service Message Definition) — 서비스 메시지 디렉토리 테이블은, 다음의 테이블에서 보여 지는 것처럼, 서비스에 사용되는 매개변수(parameter)를 정의한다:

<i>Parameter</i>	<i>Req/Ind</i>	<i>Rsp/Cnf</i>	<i>Description</i>
매개변수(parameter) X	(아래 참조)	(아래 참조)	서비스 설명.

5.3.3.1 Req/Ind 와 Rsp/Cnf 로 표현된 열은 메시지의 방향과 매개변수(parameter)를 연결한다. 개시자에 의해 전송되는 메시지는 ‘Request’로 불리며, 수신자는 이 메시지를 ‘Indication’또는 요청이라고 한다. 그리고 수신자는 ‘Response’를 전송할 수 있는데, 최초 송신자는 이를 ‘Confirmation’라고 한다.

5.3.3.2 다음의 코드는 Req/Ind 과 Rsp/Cnf 열에서 나타나고 매개 변수(parameters)의 정의에서 사용된다. (예: 각 매개 변수(parameter)가 각 방향에서 사용되는 방법):

M	필수적인 매개변수 (Mandatory Parameter) — 유효한 값을 제공해야 함.
C	조건 매개변수(Conditional Parameter): 일부 상황에서 정의될 수 있음. 값이 주어질지 말지는 완전히 선택적이며 다른 매개변수(parameter)값에 의해 정해지기도 한다.
U	사용자 정의 매개 변수 (User-Defined Parameter)
-	매개변수(parameter)가 사용되지 않음.
=	(응답에서만 사용됨) 응답에서 이 매개변수(parameter)의 값은 (만약 정의된다면) 그 초기의 매개변수(parameter)값과 반드시 일치해야 함.

6 개요 (Overview)

6.1 프로세싱 관리는 프로세싱 자원(resource)에 의한 재료의 프로세싱을 다룬다. 주된 기능(function)은 프로세싱 자원(resource)에 전달된 재료가 정확한 레시피대로 프로세스되도록 하는 것이다. 특정 재료의 프로세싱을 시작하고 추적하기 위해 관리자(서비스 사용자)가 필요로하는 서비스를 정의한다 또한 프로세싱 운영에 영향을 주는 명령(commands)을 정의한다.

6.1.1 프로세싱 자원(resource)은 재료에 제조 가치를 더하는 개체로서, 클러스터 도구 프로세싱 모듈(cluster tool process module)의 프로세싱 요소와, 완전한 독립 장비에 대한 처리를 관리하는개체뿐만 아니라 여러 형태를 가질 수 있다. 프로세싱 에이전트는 프로세싱 서비스의 제공자라고 볼 수 있다

6.1.2 프로세싱 관리는 재료의 도착 전에 ‘전 처리’(pre-conditioning)과 재료의 출발 후에 ‘후 처리’(post-conditioning)를 허가한다. 단일 튜닝(tuning) 메카니즘은 제한된 피드포워드(feedforward)와 피드백(feedback) 컨트롤(control)을 지원하며, 프로세싱이 시작될 때 적용되는 튜닝(tuning)은 레시피 가변 매개변수(parameter)를 설정한다.

6.1.3 서비스는 프로세싱 에이전트(서비스-제공자)가 제공하는 모든 기능을 정의하며, 관리자(서비스-사용자)의 아키텍처(architecture)를 기술하지는 않는다.

6.1.4 이 표준은 통신이 기반이 되는 상황에서 개념과 프로세싱 모델을 정의하고, 뒤이어 사용되는 동작 모델의 자세한 설명이 따른다. 더불어 표준 객체 속성과 메시지 서비스들이 자세하게 설명된다.

6.2 표준 준수(Compliance) — 표준에 대한 준수사항은 이 문서에서 명시된 모든 요구사항의 준수를 의미한다. 관련된 기능이 요구되는 경우 표준 서비스가 사용된다. 정의된 메시지 서비스와 상태모델들이 이에 포함된다

6.2.1 대기(queueing), 다중 병렬 작업(jobs), 재료 그룹들, 매뉴얼 시작, 중단/재시작, 튜닝과 같은, 일부 기능들은 표준 준수를 위해 반드시 지원되어야 할 요구사항은 아니다. 필수적인 기능들은 이 문서 전체에 걸쳐 명시되고 또한 필수적인 요구사항(Fundamental Requirements)에서 나열된다.

6.2.2 프로세싱 에이전트는 필수 요구 사항을 제공해야만하며 추가적으로 특정한 하드웨어 구조와 자동화 프로세싱 요구사항에 대한 효과적인 프로세싱 관리를 위해 적절한 선택적 요구사항도 제공한다.

7 개념 (Concepts)

7.1 재료 프로세싱 모델 (Material Processing Model) — 프로세싱 관리는 프로세스 잡(process job) 정의를 통한 적절한 프로세싱이 프로세싱 자원(resource)에 의해 특정 재료에 적용되도록 한다. 프로세스 잡(process job)은 사용되고 있는 특정 프로세스가 무엇이던간에, 장비 내에서 재료의 자동 처리를 위해 널리 적용할 수 있는 관리 컨트롤(control) 기능을 제공한다.

7.1.1 이 표준(Standard)은 재료와 레시피 사양이 주어진다면, 프로세싱 자원(resource)이 요구된 프로세싱의 목적을 독립적으로 달성 할 수 있다고 가정한다.

7.1.2 프로세싱 관리는 재료 이동에 대한 서비스를 제공하지는 않지만, 서비스 제공자가 재료의 수신과 전송에 관한 활동을 조정할 필요가 있으며 이를 통하여 시스템 완전성을 유지할 수 있게 된다.

7.2 프로세스 잡 (Process Job) — 프로세스 잡(process job)은 프로세싱 자원(resource)으로 재료의 프로세싱을 실행하기 위해 프로세싱 관리자(서비스 사용자)에 의해 명시된 활동적인 객체이다. 상위 단계 작업(job)은 일단 재료가 도착하면, 관리자의 추가 중재 없이 재료의 프로세싱이 이뤄지도록 프로세싱 자원(resource)이 요구하는 모든 정보를 포함한다.

7.2.1 프로세스 잡(process job)은 네 가지 순차적인 단계를 포함한다:

- 재료 도착 전 프로세싱 자원(resource)의 전 처리
- 프로세싱을 위해 재료와 프로세싱 자원(resource) 준비
- 재료 프로세싱
- 재료의 출발(departure) 후 프로세싱 자원(resource) 후 처리.

7.2.2 재료 프로세싱 단계는 재료가 변경되는 경우에만 요구되는 유일한 단계이다.

7.2.3 이 표준(Standard)은 프로세스 잡(process job)의 생성, 컨트롤(중단(pausing), 중지(aborting) 등) 및, 트래킹(tracking)을 위한 서비스를 명시한다. 이는 어플리케이션에 의존적(application-dependent)이기 때문에 프로세싱의 낮은 수준의 컨트롤(control)은 정의하지 않는다. 프로세스 잡(process job)을 수행하는 프로세싱 자원(resource)은 프로세싱 목적을 이루기 위해 적절하다고 간주되는 것은 무엇이던지 레시피와 튜닝 매개변수(parameters)에 명시된대로 수행해야할 책임이 있다.

7.2.4 프로세스 잡(process job)에서 명시된 재료는 실제 처리되는 단일 재료 요소이며 웨이퍼의 카세트와 같은 컨테이너들이 이에 해당 된다.

7.2.5 프로세스 잡(process job)의 수명주기(lifecycle)는 재료에 실행되는 프로세스 범위를 넘어 연장될 수 있으며, 설정(setup)과 프로세싱을 통해 재료 도착 전부터 재료 출발 후까지를 말한다. 이는 재료가 도착하기 전 프로세싱 자원(resource)의 재료 프로세싱과 관련한 전처리와 재료가 보내진 후 프로세싱 자원(resource)의 후 처리를 (예, 청소) 허용한다. 전 처리와 후 처리의 지원은 필수적 요구사항은 아니다.

7.2.6 프로세싱 자원(resource)은 작업이 미리 계획되어 있거나 재료의 도착 순서가 정해지지 않은 시스템에 유연성을 제공하기 위해 프로세스 잡(process job) 대기를 제공한다. 대기는 프로세싱 활동에 앞서 다중 프로세스 잡 (process jobs)의 수락을 의미한다. 일반적으로 동시 작업 및 연속 작업 (job)을 요청하는 보다 복잡한 시스템에서 이 기능이 지원될 필요가 있다(아래 참조). 작업(jobs)은 생성된 순서로 대기열에 나열되며, 동일한 재료의 연속적인 작업(jobs)을 수행하는 경우에는 실행순서가 중요하기도 하다. 대기는 필수 요구사항은 아니다.

7.3 재료 관련 이동 (*Relation to Material Movement*) — 프로세싱 관리(Processing Management)는 프로세싱을 위해 프로세싱 에이전트 도메인으로 재료를 수신하거나 프로세싱이 완료된 후 재료를 보내는 서비스를 제공 하지 않는다.

7.3.1 프로세싱은 재료의 존재여부에 따라, 재료의 출발은 프로세싱 완료 여부에 따라 달라진다. 프로세싱 자원(resource)의 전 처리와 후 처리가 적용된다면, 재료 이동의 동기화가 필요한 상호 의존성이 생긴다. 장비는 재료 전송과 프로세싱 간에 완전성(integrity)을 유지할 책임이 있다.

7.3.2 재료 이동 관리는 이 표준의 범위를 벗어나지만 적용 가능한 SEMI 표준을 적용 함으로써 수행될 수 있다.

7.4 프로세싱 설명 (*Processing Description*) — 프로세스 잡(process job)에 의해 적용되는 프로세싱의 설명은 매우 중요하다. 이 설명은 프로세싱 레시피 (Process Recipe)의 (SEMI E42 참고) 형태나 프로세싱 프로그램 (Process Program) (SEMI E30 참조)의 형태로 제공될 수 있다. 이 사양은 프로세싱 레시피(Process Recipes)만을 언급하는 메시지를 정의하며, 프로세스 레시피 (Process Recipe) 대신 프로세스 프로그램(Process Programs)을 사용할 것을 특별히 고려해야할 경우에는 이 내용이 따로 언급된다.

7.4.1 프로세스 잡(Process Job)은 프로세싱 에이전트의 도메인 내에서 고유한 프로세싱의 설명 (레시피 또는 프로세스 프로그램) 식별자를 포함한다. 프로세싱 설명의 유형과 내용은 프로세싱 자원(resource)과 재료의 유형에 적용 가능해야 한다.

7.4.2 레시피와 프로세스 프로그램의 생성과 관리는 이 표준의 범위를 벗어난다.

7.5 프로세스 튜닝 (*Process Tuning*) — 프로세스 단계 사이의 피드포워드(feedforward)와 피드백(feedback) 컨트롤(control)은 현장 계측 장비의 부족과 수율(혹은 성능)의 향상과 프로세스 안정화를 위해 요구되는 프로세스 튜닝(tuning)을 위해 더욱 중요해지고 있다. 이 프로세스 튜닝의 요구사항은 응용 프로그램과는 상당히 다르지만, 특정한 메소드(method)에 대해서는 다소 의견이 일치한다. 프로세스 튜닝에 대한

종합적인 지원을 제공하는 것을 이 표준이 의도하는 것이 아니라 확장될 수 있는 단순한 메커니즘을 제공하는 것이 목적이다. 레시피 튜닝에 대한 지원은 필수적 요구사항은 아니다.

7.5.1 프로세싱 관리는 적용될 레시피 메소드(method)의 사양에 대한 메커니즘을 제공하며, 두 가지 메소드(methods)가 표준에서 정의된다: RecipeID only, RecipeID 와 변수(Variables). 또한 특정한 정의와 응용 프로그램의 요구사항에 대한 일반적 이해를 가지는 모든 통신 개체를 요구하기 때문에 사용자-정의 메소드(methods)가 사용되기도 한다.

7.5.2 RecipeID 는 추가적인 튜닝 매개변수(parameters)를 제외한 적용되는 레시피의 식별자를 수용하는 유일한 메소드다. 응용 프로그램의 레시피 바디가 응용 프로그램의 튜닝을 정의하는 것을 막지 않지만 표준화된 지원은 없다.

7.5.3 RecipeID 와 변수 메소드(Variables Method)는 피드포워드(feedforward)와 피드백(feedback) 컨트롤(control)을 운영하기 위해 제한된 운용을 지원하는 프로세스 잡(process job) 생성시 단순한 프로세스 튜닝 메커니즘을 제공한다. 이는 변수 튜닝 메소드(VariableTuning method)를 정의하고 프로세스 잡 생성(process job create) 에서 변수 이름과 값 리스트를 제공하며 이를 통해 레시피에서 정의된 변수들을 설정한다. 각 변수의 이름은 레시피 관리에서 지원된 노출 변수 정의 중에 하나이며, 해당 값은 변수 정의에서 명시된 범위 내에 들어갈 것이다.

7.5.4 레시피 매개변수 명칭(RecipeVarName)은 SEMI40 의 범위 내에서 SEMI E39 (Object Services Standard)의 ‘Object Specification’ (ObjSpec)에서 정의한 명명법을 이용하여 명시된다 (Note: SEMI E5 의 OBJSPEC 참조). ObjSpec 명명법은 복잡한 레시피(예, 클러스터 도구 프로세스 레시피) 내에서 매개변수(parameter)를 명확하게 식별하기 위해 사용된다. 레시피 매개변수(parameter)의 사양이 명확하다면, ObjSpec 의 형태는 단지 매개변수(parameter)의 명칭으로 단순화 될 것이다. SEMI E40 레시피 매개변수(parameter) 명칭을 위한 ObjSpec 명명법의 사용은 장비 레시피 구조를 설명하는 객체 모델을 요구한다.

7.5.5 Figure 1 은 클러스터 도구 계층 레시피 관계(cluster tool hierarchical relationships)의 집합의 전형적인 예이다. 프로세싱 매개변수(parameter)는 시퀀스 단계(Sequence Step) AB, 프로세스 레시피(Process Recipe) BB, 프로세스 단계(Process Step) CB, 프로세스 매개변수(Process Parameter)DB 를 통해 명시된다. 이 예에서, 프로세스 매개변수 DB (Process ParameterDB)를 위한 객체 규제자(object specifier)는:

“Sequence:Erma>SequenceStep:AB>ProcessRecipe:
BB>ProcessStep:CB>ProcessParameter:DB>”

혹은

“Erma>AB>BB>CB>DB>”

이다.

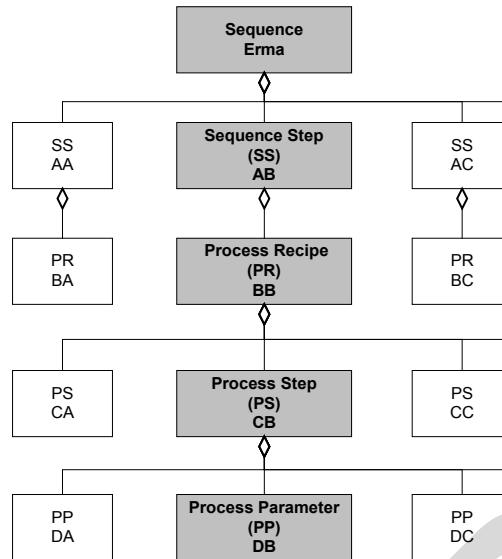


Figure 1
클러스터 도구 시퀀스 레시피 구조 관계
(Cluster Tool Sequence Recipe Hierarchical Relationships)

7.6 프로세싱 재료 그룹 (*Processing Material Groups*) — 대다수의 장비 구조는 재료 그룹에 대해 동시 프로세싱을 요구한다. 단일 프로세스 잡(process job)은 특정 제한을 가지는 재료 그룹을 컨트롤(control)할 수 있고, 그룹 내의 모든 재료는 동일한 유형을 가지고 동일하게 처리 될것이 요구된다. 각 재료의 프로세싱은 그룹의 도착에 따라 달라지는데, 즉 전체 재료 그룹은 프로세싱의 시작 전 장비에 도착하고, 모든 프로세싱이 완료될 때까지 이동하지 않는다. 프로세스 잡(process job)은 단순한 논리적 통제 메커니즘을 유지하는 반면 강력한 통제하에 재료 이동은 원활해야 하며 재료 데이터의 추적이 효과적이어야 한다.

7.6.1 허용되는 재료 그룹 프로세싱의 두 가지 일반적인 예:

- 동일하게 처리되는 웨이퍼의 카세트를 수신하는 장비, 그룹 내의 모든 웨이퍼는 카세트 담겨 수신된다. 웨이퍼의 프로세싱이 동시 진행되거나 아닐 수도 있지만 일단 카세트가 도착해야만 진행할 수 있다. 모든 웨이퍼 프로세싱은 카세트가 제거 되기 전에 완료된다. 카세트를 재료로 간주하여 단순 프로세스 잡을 명시하기도 한다는 것을 유의한다
- 웨이퍼를 함께 처리하는 클러스터 도구 배치(batch)는 모듈. 웨이퍼는 단독으로 수신되며, 프로세스 잡(process job)에 명시된 모든 웨이퍼가 도착할 때 배치(batch) 챔버에서 웨이퍼 그룹의 프로세싱이 시작된다. 웨이퍼는 프로세싱이 완료된 후 전송된다.

7.7 병렬 프로세스 잡(*Concurrent Process Jobs*) — 병렬 프로세스 잡(process jobs)은 동시에 다중의 작업(multiple jobs)이 (대기없이) 실행되는 상황을 말한다. 다중 재료의 병렬 프로세싱에 단일 프로세스 잡(process job)이 항상 적합한 것은 아니다. 이런 경우, 다중 병렬 작업(jobs)이 프로세싱 자원(resource)에 의해 지원되고, 병렬 프로세스 잡(process jobs)의 지원은 필수적인 요구사항이 아니다.