SEMI E174-1116
SPECIFICATION FOR WAFER JOB MANAGEMENT (WJM)

This Standard was technically approved by the Information & Control Global Technical Committee. This edition was approved for publication by the global Audits and Reviews Subcommittee on October 17, 2016. Available at www.semiviews.org and www.semi.org in November 2016.

1 Purpose

1.1 The purpose of this Standard is to provide a communication method which observes and/or controls every single wafer (including its related information) on equipment individually (and independently rather than a member of a lot) through single management point from the host.

1.2 The purpose of this Standard is to introduce Wafer Job (WJ) in order to observe and/or control a wafer (and its information including context information from related actions such as pre-processing, processing, and post-processing steps) in the equipment independently.

1.3 The purpose of this Standard is to provide Wafer Job Object (WJOBJ) model so that the information that should be managed in conjunction with the WJ can be encapsulated in the object and can be accessed from the host.

1.4 The purpose of this Standard is to introduce Wafer Flow Job (WFJ) in order to manage multiple WJs in the equipment.

1.5 The purpose of this Standard is to provide Wafer Flow Job Object (WFJOBJ) model so that the information that should be managed in conjunction with the WFJ can be encapsulated in the object and can be accessed from the host.

2 Scope

2.1 The scope of this Standard is to define WJOBJ which represents WJ.

2.2 The scope of this Standard is to define WFJOBJ which represents WFJ.

2.3 The scope of this Standard is to define related functionalities which support WJ and/or WFJ.

2.4 This Standard mainly focuses on product wafer, however does not exclude non-product wafers.

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.

3 Limitations

3.1 *Commonly Required Specifications Only* — This Standard only provides commonly required specifications to create, maintain, and delete a WJOBJ and a WFJOBJ by equipment, and to observe and/or control a wafer by the host via the WJOBJ and WFJOBJ. For equipment dependent operations, user of this Standard should define required attributes to support those operations.

4 Referenced Standards and Documents

4.1 *SEMI Standards and Safety Guidelines*

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

SEMI E39 — Object Services Standard: Concepts, Behavior, and Services (OSS)

SEMI E40 — Standard for Processing Management (PJM)

SEMI E40.1 — SECS-II Support for Processing Management Standard

SEMI E87 — Specification for Carrier Management (CMS)

SEMI E90 — Specification for Substrate Tracking (STS)

SEMI E94 — Specification for Control Job Management (CJM)

SUPERSEDED

Documents Document Sharing Hub

Full document details: SEMI E174-1116

SEMI E142 — Specification for Substrate Mapping

SEMI E157 — Specification for Module Process Tracking

SEMI E170 — Specification for Secured Foundation Of Recipe Management System (SFORMS)

SEMI T7 — Specification for Back Surface Marking of Double-Side Polished Wafers with a Two-Dimensional Matrix Code Symbol

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

Table 1 Referenced Attributes and Variables

Name	*Defined in*[#1]	*Description*[#2]	*Form*[#2]
PRJobID	SEMI E40 PJM	The unique identifier for a process job. It can be accessed as the ObjID attribute of the process job. The host may provide this identifier to the processing resource. In this case, the host must guarantee uniqueness of job identifiers for all the process jobs in the PROCESS JOB STATE in the equipment.	Text
PRRecipeMethod	SEMI E40 PJM	Indication of recipe specification type, whether using is applied and which method is used.	Enumerated: Recipe only Recipe with Variable Tuning
RecID	SEMI E40 PJM	Identifier of recipe applied.	Text
RecVariableList	SEMI E40 PJM	List of variables supporting a recipe method.	List of RecipeVariable RecipeVarilable is a structure composed of: RecipeVarName, RecipeVarValue
PRMtlName	SEMI E40 PJM	Textual identifier of the material being processed.	Text: Unique for each material with respect to the processing agent.
PRMtlNameList	SEMI E40 PJM	List of identifiers of the material being processed.	List of: PRMtlName OR Structure composed of: Carrier IDs and corresponding Slot IDs. The format of the structure is specified in the implementation standard used. The SECS-II structure format is specified in SEMI E40.1.
Slot ID	SEMI E40 PJM	Slot identifier.	
Carrier ID	SEMI E40 PJM SEMI E94 CJM	Identifier of wafer which is a combination of Carrier ID and SlotID.	See actual structure for each Standard which uses this combination.
CtrlJobID	SEMI E94 CJM	Control job identifier, available to be used in control job related event reports.	Text
MtrlOutSpec	SEMI E94 CJM	Maps material from source to destination after processing. In Slot Integrity Mode the list shall be empty.	List of structure: SourceMap DestinationMap
SubstrateLocation	SEMI E94 CJM	A substrate position at a source and a destination. A carrier is an example of a multi-location destination. For a wafer carrier the SubstrationLocation is a slot number.	Numeric

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E174-1116

Name	*Defined in*[#1]	*Description*[#2]	*Form*[#2]
CarrierID	SEMI E87 CMS	Identifier of carrier.	Text Conforms to ObjID as defined in SEMI E39.
ObjID (of SEMI E90 STS)	SEMI E90 STS	Object identifier.	Text equal to the Substrate ID.
AcquiredID	SEMI E90 STS	Contains the ID read from the substrate. Empty string before the substrate is read. The attribute shall be updated as soon as the substrate ID has been successfully read. [Equipment with substrate ID reader only]	Main purpose of this is for exception check and/or traceability in case wafer is not under control of fab system. Text
SubstID	SEMI E90 STS	Substrate Identifier relevant to the location.	Text
Default ID (Not as an attribute. Defined in terminology)	SEMI E90 STS	The substrate ID assigned to the substrate when no substrate ID information is given by the user but the carrier ID for the source carrier is known.	The default ID is the combined text of the source carrier ID and the slot number.
SubstLocID	SEMI E90 STS	Identifier for current equipment substrate location or carrier substrate location that the substrate occupies.	Text (If batch process equipment, this data is empty while the substrate belongs to a batch.)
BatchLocID	SEMI E90 STS	Identifier for batch location associated with the last event.	Text
SubstPosInBatch	SEMI E90 STS	Current substrate position in a batch.	Text (Batch process equipment only, and empty if substrate position in a batch is not informed.)

#1 See original Standard for definition.

#2 Quotations from referenced Standards. In case differences are found, definitions in the referenced Standards precede the quotations here.

5 Terminology

5.1 *Abbreviations and Acronyms*

5.1.1 *EXG* — execution group

5.1.2 *GEM* — generic equipment model

5.1.3 *PJ* — process job

5.1.4 *PJG* — process job group

5.1.5 *SPOWM* — single point of wafer management

5.1.6 *VM* — virtual metrology

5.1.7 *VP* — variable parameter

5.1.8 *WFJ* — Wafer Flow Job

5.1.9 *WFJOBJ* — Wafer Flow Job Object

5.1.10 *WJ* — Wafer Job

5.1.11 *WJOBJ* — Wafer Job Object

5.1.12 *WOM* — wafer oriented management

5.2 *Definitions*

5.2.1 *buffer* — a set of one or more locations for holding carriers at/inside the production equipment.

5.2.2 *carrier* — a container, such as a FOUP or open cassette, with one or more positions for holding substrates.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E174-1116

5.2.3 *CarrierID* — a readable and unique identifier for the carrier.

5.2.4 *collection event* — a collection event is an event (or grouping of related events) on the equipment that is considered to be significant to the host.

5.2.5 *host* — the factory computer system or an intermediate system that represents the factory and the user to the equipment.

5.2.6 *load* — the operation of placing a carrier on a load port.

5.2.7 *load port* — the interface location on the equipment where carriers are loaded and unloaded.

5.2.8 *post-processing* — a neutralization action which is done by a process means after a wafer is unloaded from the process means such as post-conditioning of a chamber with gas flow.

5.2.9 *pre-processing* — an action which is done by a process means before a wafer is loaded to the process means such as preconditioning of the chamber with a gas flow, preheating of the chamber or hot plate.

5.2.10 *execution group (EXG)* — a group of wafer which is dealt at a time in the same environment by using the same control condition.

5.2.11 *process job group (PJG)* — a group of wafer which is specified by one PJ.

5.2.12 *production equipment* — equipment used to produce product, such as semiconductor devices, including substrate sorting, process, and metrology equipment and excluding material handling equipment.

5.2.13 *properties* — a set of name value pairs assigned to an object or used in a service message to include additional information about the object (i.e., carrier, port, etc.).

5.2.14 *slot map* — the information that relates which slots in a carrier hold substrates, both correctly and incorrectly.

5.2.15 *substrate* — material held within a carrier. This can be product, or durables such as reticles.

5.2.16 *unload* — the operation of removing a carrier from a load port.

5.2.17 *user* — a human or humans who represent the factory and enforce the factory operation model. A user is considered to be responsible for many setup and configuration activities that cause the equipment to best conform to factory operations practices. [SEMI E30]

5.2.18 *virtual metrology (VM)* — a method which provides estimated metrology variables of a process result of the object from its control conditions and its indirect measurement values by using a modeling technique.

5.2.19 *Wafer Flow Job (WFJ)* — a management capability which observes and/or controls multiple WJs in the equipment.

5.2.20 *Wafer Flow Job Object (WFJOBJ)* — a software representation of a wafer flow job in the equipment.

5.2.21 *Wafer Job (WJ)* — a management capability which observes and/or controls a wafer in the equipment independently.

5.2.22 *wafer job flow (WJFlow)* — a list of WJs which are in a same flow in wafer management.

5.2.23 *Wafer Job Object (WJOBJ)* — a software representation of a wafer job in the equipment.

6 Conventions

6.1 *Objects*

6.1.1 Whenever the equipment is required to know about specific kinds of entities and required to manage information concerning these entities, it is useful to treat these entities as objects that comply with the basic requirements of SEMI E39, Object Services Standard (OSS). This is especially true whenever there are a large number of objects of a given type or when the entities are transient rather than permanent. In both cases, it is difficult to describe a general way for the host and equipment to specify which particular entity is referenced and to get information related only to a specific one out of many.

6.1.2 By defining these entities as objects that comply with OSS, it is only necessary for the host to specify the type of object and its specific identifier in order to inquire about one or more properties of the specific entity of interest.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E174-1116

6.1.3 *Object Properties*

6.1.3.1 A property (attribute) is information about an individual object that is presented as a name and value pair. The name is a formally reserved text string that represents the property, and the value is the current setting for that property.

6.1.3.2 Properties shall be accessible to the host via the service GetAttr. Using SEMI E39 Object Services Standard, for example, it is possible to:

- get the list of IDs for the current objects at the equipment, and
- get the specified properties for one or more individual objects.

6.1.4 *Rules for Object Properties*

- Attributes with RO (Read Only) access cannot be changed using SetAttr service as defined in OSS.
- Attributes with RW (Read/Write) access can be changed using SetAttr service as defined in OSS.
- Additional attributes may be specified by the user or the equipment supplier by using an attribute name starting with User Defined (UD). Care should be taken to ensure the name of the attribute is unique.

6.1.5 *Object Attribute Table*

6.1.5.1 The object attribute table is used to list all the attributes related to the defined object as shown below. The access is defined as Read Only (RO) or Read/Write (RW). The Reqd column is used to specify whether the attribute is required (Y), conditional (C), or optional (O) for implementation. Finally, the Form column is used to specify the format of that particular attribute.

Table 2 Object Attribute Table

Attribute Name	*Definition*	*Access*	*Reqd*	*Form*

6.2 *State Model Methodology*

6.2.1 A state model consists of four elements: a State Model Diagram, a State Model Definition Table, a State Definition, and a State Transition Table.

6.2.2 *State Model Diagram* — The diagram of the state model uses the Harel State Chart notation. An overview of this notation is presented in an Appendix of SEMI E30. The definition of this notation is presented in Science of Computer Programming 8, "Statecharts: A Visual Formalism for Complex Systems," by D. Harel, 1987.[1]

6.2.3 *State Model Definition Table* — The State Model Definition Table used in this Standard has the following format. This table defines states and possible transition(s) from each state side by side. Each state has one or more transitions. When the transition comes from outside this table, the state definition column associated with the transition may be blank (see [#1]). When the transition comes from unspecified multiple states with the same condition, the state definition column may say '#Any state', and there may not be an explicit transition number.

6.2.3.1 *Definition of State* — Columns under the State column define States with No. (Number), Name and Abstract of Definition. 'No.' corresponds to the state number in the associated state diagram. 'Name' defines a name of each state. 'Abstract of Definition' provides an abstract of the State Definition in the State Definition Table.

6.2.3.2 *Definition of Transition* — Columns under the Transition column define Transitions with No. (Number), Abstract of Trigger, Abstract of Action, New State, and Comments. 'No.' corresponds to transition number in the associated state diagram. 'Abstract of Trigger' and 'Abstract of Action' provide abstracts of the Trigger and Action in the State Transition Table accordingly. 'New State' defines a state number to move to after the transition is completed by pointing to one of the states defined in the state definition in left side of the table. The Comment column may be used to put comments to each transition or their From state.

[1] Elsevier Science, P. O. Box 945, New York, NY 10159-0945; http://www.journals.elsevier.com/science-of-computer-programming/

SUPERSEDED
Documents Document Sharing Hub
Full document details: SEMI E174-1116

Table 3 State Model Definition Table

State			*Transition*				
No.	*Name*	*Abstract of Definition*	*No.*	*Abstract of Trigger*	*Abstract of Action*	*New State*	*Comments*
-#1	-#1	-#1	T00			S00	
Snn#2	# Any state#2	-#2	-			S03	
S00			T01			S01	
S04#3			T02			S00	
S01			T03			S02	
			T04			S03	
S02			T05#4			S00	
S10#5			-	-	-	-	
S11			T10			S12	
S12			T11			S11	
S03			T06			S00	

#1 When the transition comes from outside of this table, the state definition column may be blank.

#3 When the transition comes from unspecified multiple states with the same condition, the state definition column may say '#Any state'.

#4 A state which has substates.

#5 One transition path has multiple cases (trigger and action pairs).

#6 A state separated by a dotted line is a parallel state of the state above.

6.2.4 *State Definition Table* — State definition tables are provided in conjunction with the state diagrams to explicitly describe the definition of each state. A state definition table contains columns for State Number, Mnemonic, State Definition, and Comments.

Table 4 State Definition Table

No.	*Mnemonic*	*State Definition*	*Comments*

6.2.5 *State Transition Table* — State Transition Tables are provided in conjunction with the state diagrams to explicitly describe the nature of each state transition. A State Transition Table contains columns for Transition Number, Previous State, Trigger, New State, Actions, and Comments. The 'trigger' (column 3) for the transition occurs while in the Previous State. The 'Actions' (column 5) includes a combination of:

- Actions taken upon exit of the previous state,
- Actions taken upon entry of the new state, and
- Actions taken which are most closely associated with the transition.

Table 5 State Transition Table

No.	*Previous State*	*Trigger*	*New State*	*Actions*	*Comments*

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E174-1116

6.2.6 *State Model Requirements*

6.2.6.1 *Requirement* — The state models included in this Standard are a requirement for compliance. Equipment shall maintain state models for each of the required state models as defined in this Standard. Equipment shall maintain individual and unique state models for each logical entity instantiated or each physical entity in the equipment that has associated state models.

6.2.6.2 *Representation as the Host View* — A state model represents the host's view of the equipment, and does not necessarily describe the internal equipment operation. All state model transitions shall be mapped sequentially into the appropriate internal equipment collection events that satisfy the requirements of those transitions. In certain implementations, the equipment may enter a state and have already satisfied all of the conditions required by the state models in this Standard for transition to another state. In this case, the equipment makes the required transition without any additional actions.

6.2.6.3 *Additional Sub-States* — Some equipment may need to include additional substates other than those in this Standard. Additional substates may be added, but shall not change the defined state transitions in this Standard. All expected transitions between states in this Standard shall occur.

6.2.6.4 *Uniqueness of Event Identifier* — The event identifier reported during a particular state transition change for each of these state models shall be shared for all associated state models but unique for each transition. For example, if the equipment has two load ports and the load port state model defines ten transitions, there must be exactly ten event identifiers for each load port transfer state model but not ten for each physical load port. The information identifying the physical entity or logical entity undergoing the transition will be contained within the associated event report.

6.2.6.5 *Events* — All state transitions in this Standard, unless otherwise specified, shall correspond to collection events. More explicitly, there must be a unique collection event for each state transition.

6.2.6.6 *Events for Multiple AND Substates* — When a state model is defined with multiple AND substates, the equipment may report all state entry events with only one collection event.

6.2.6.7 *Events for Conditional Path* — When conditional paths are defined in the state model, it is not necessary to report any state transition(s) until a terminal state is reached at which time each transition used to reach that state is reported.

6.3 *Object*

6.3.1 *Recognition of Object* — From the host point of view, an object is instantiated if the host is able to query the equipment about that object, its current state, and other attributes. Once instantiated, the object is considered destroyed (no longer instantiated) if the response to such queries is 'unknown object'.

6.3.2 *Object Identifier (ObjID)* — The purpose of an Object Identifier is to allow references to an object within the system. The Object Identifier is assigned when an object is instantiated and should be unchanged or persistent until the end of the object lifecycle. The Object Identifier shall be unique at the equipment during lifecycle of the object.

6.4 *Services*

6.4.1 *Definition of Service* — Services are functions or methods that may be provided by either the equipment or the host. A service message may be either a request message, which always requires a response, or a notification message that does not require a response.

6.4.1.1 *Notification Message Service* — Notification type messages are initiated by the service provider (e.g., the equipment) and the provider does not expect to get a response from the service user.

6.4.1.2 *Request Message Service* — Request messages are initiated by a service user (e.g., the host). Request messages ask for data or an activity from the provider. Request messages expect a specific response message (no presumption on the message content).

6.4.2 *Service Message Description* — A service message description table defines the parameters used in a service, as shown in the following table:

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E174-1116

Table 6 Service Message Description Table

Service Name	*Triggered by*	*Type#1*	*Req#2*	*Description*

#1 The 'Type' column is used to indicate whether the service consists of a request/response message pair, 'R', or a single notification message, 'N'.

#2 The 'Req' column is used to indicate whether the service is required, 'R', or optional, 'O'.

6.4.3 *Service Message Parameter Definition* — A service parameter dictionary table defines the description, range, and type for parameters used by services, as shown in the following table:

Table 7 Service Message Parameter Definition Table

Parameter Name	*Form*	*Description*
#1		

#1 A row is provided in the table for each parameter used on a service.

6.4.4 *Service Message Definition* — A service message description table defines the parameters used in a service message, and describes each message and its cause/effect to the equipment, as shown in the following table:

Table 8 Service Message Definition Table

Service Parameter	*Req/Ind*	*Rsp/Conf*	*Description*

6.4.4.1 *Definition of Req/Ind and Rsp/Conf Columns* — The columns labeled Req/Ind and Rsp/Conf link the parameters to the direction of the message. The message sent by the initiator is called the 'Request'. The receiver terms this message the 'Indication'. The receiver may then send a 'Response', which the original sender terms the 'Confirmation'.

6.4.4.2 *Definition of Codes for Req/Ind and Rsp/Conf Columns* — The following codes appear in the Req/Ind and Rsp/Conf columns and are used in the definition of the parameters (i.e., how each parameter is used in each direction):

Table 9 Codes for Req/Ind and Rsp/Conf Columns

M	Mandatory Parameter – must be given a valid value
C	Conditional Parameter – may be defined in some circumstances and undefined in others. Whether a value is given may be completely optional or may depend on the values of other parameters.
U	User-Defined Parameter
-	The parameter is not used
=	(for response only) Indicates that the value of this parameter in the response must match that in the primary (if defined)

6.5 *Variable Data Definitions*

6.5.1 Variable data definitions define variable data requirements. Values of these variables are available to the host via collection event reports and host status queries.

6.5.2 *Event Report Requirement* — The identifier of an object and all of the attributes of that object shall be available for inclusion in event reports associated with that object.

6.5.3 *Object Attribute Variable in Non-Extinction Event* — The object attribute variables in event reports linked to non-extinction event(s) shall contain the values of the attributes after the transition. This requirement allows the receiver of the report to know the current condition of the object.

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E174-1116

6.5.4 *Object Attribute Variable in Extinction Event* — The object attribute variables in event reports linked to extinction event(s) shall contain the values of the attributes before the transition unless it is specifically stated that the destruction transition modifies the attribute value. This requirement allows the receiver of the report to know the final condition of the object at the time it was deleted.

6.5.5 Subscripted variables are used either as items within a list or to differentiate data representing different entities.

6.5.6 *Table Format* — The following table defines variable data that shall be provided by the production equipment.

Table 10 Variable Data Definitions

Variable Name	*Description*	*Type*	*Access*	*Comment*

7 Conceptual Description

7.1 *Conceptual Description* — This section describes the concept of WJM, and includes no requirement.

7.2 *Issues and Requirements to be Addressed* — This Standard is developed to be compliant with the issues and requirements such as:

- Easier and quicker context data gathering of individual wafer for quick process/environment analysis.
- More precise production control of every single wafer.
- Dynamic modification of process conditions such as Variable Parameters per wafer by W2W APC.
- Flexible lot split or merge on equipment.
- Diagnostic use of behavior tracking data from individual wafer (including non-product or even jig wafers which are not fully controlled by the host).

7.3 *Major Concepts* — In order to solve the issues and to be compliant with the requirements, this Standard introduces the following major concepts.

7.3.1 *Single Point of Wafer Management (SPOWM)* — A capability which observes and/or controls wafers (including its related information) on equipment from the host through single management point called WFJ.

7.3.2 *Wafer Oriented Management (WOM)* — A capability which manages every single wafer independently by WJ rather than a member of a lot (or PJ).

7.3.3 *Coverage of Wafer Types* — This Standard primarily covers product wafers under the production management of the host, then covers non-product wafers under the production management of the host, and also, recommends to cover wafers which are locally managed in equipment and not under production management of the host.

7.4 *Introduction of WJ* — In order to provide a capability to observe and/or control a wafer in the equipment individually from the host, this Standard introduces WJ to manage the wafer.

7.4.1 *Relationship with Factory WJ* — WJ is valid only in the equipment in order to make the equipment design independent from the host design. Correlation between equipment managed WJ and factory WJ should be maintained by factory side. To help this, WJOBJ should have an attribute which holds WJ identification in the host side.

7.4.2 *Life of WJ* — WJ takes place during a wafer or its information exists in the equipment.

7.5 *Introduction of WJOBJ* — In order to access to a WJ in the equipment individually from the host, this Standard introduces WJOBJ as a representation of a WJ.

7.5.1 *Who to Create and Delete a WJOBJ* — WJOBJ is created and deleted by the equipment.

7.5.1.1 *Creation of WJOBJ* — WJOBJ is created no later than the wafer or information of the wafer that should be managed by the WJ arrives to the equipment.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E174-1116

7.5.1.2 *Deletion of WJOBJ* — WJOBJ is deleted just after the wafer is removed from the equipment and its information is read out from the object.

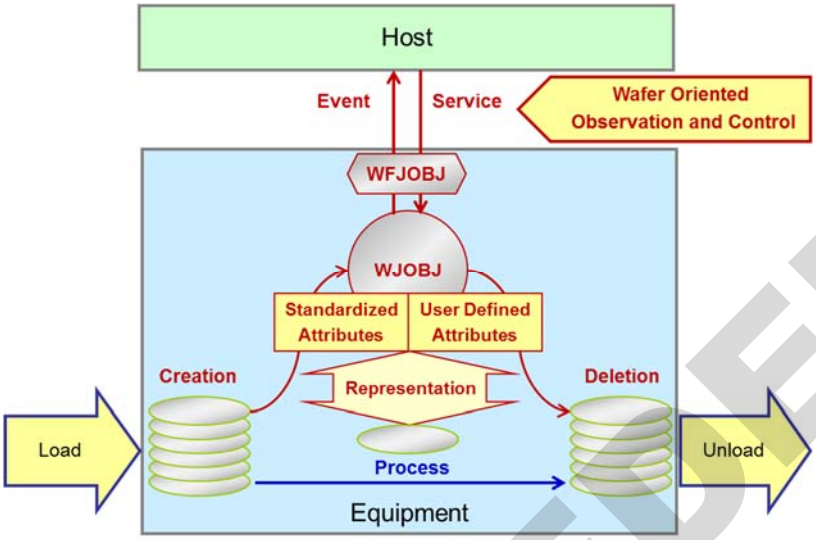

Figure 1
WJOBJ, WFJOBJ and Wafer Level Control in Equipment

7.5.2 *Attributes of WJOBJ* — The following attribute structure concept is introduced.

- *Main Attributes* — Main attributes mainly cover following information:
  - Identifiers of WJOBJ itself
  - State of the WJOBJ
  - Identifiers of a wafer the WJOBJ takes care of
- *Process Attributes* — Process attributes cover wafer process related information such as:
  - Parent PJ identifier
  - Recipe identifier inherited from the parent PJ
  - Recipe variables inherited from the parent PJ and may be modified per wafer by WJ service afterward
- *Transfer Attributes* — Transfer attributes cover wafer to carrier correlation information such as:
  - Source carrier and slot (in which carrier the wafer is transported in)
  - Destination carrier and slot which may be modified dynamically (in which carrier the wafer will be transported out)
- *Location Tracking Attributes* — Location tracking attributes provide some of the information from SEMI E90 STS to provide direct correlation to WJ.
  - Substrate location
  - Batch location (Optional)
- *Result Record Attributes* — Result record attributes hold result information to respond to read request from the host until WJOBJ is deleted.
  - Process result
  - Wafer path

SUPERSEDED

Documents
Document Sharing Hub
Full document details: SEMI E174-1116

7.5.3 *State Management of WJOBJ* — This Standard provides WJOBJ State Model in order to show in which job state the WJ is. States of WJ consists of non-equipment-specific states and equipment-specific states and is not needed to model every small state changes which is not needed to be notified to the host. WJM provides a simple and scalable state model, and equipment to define actual states and transitions important to perform its own wafer process in communication with the host. This Standard only provides the Meta model of the WJOBJ State Model, since State Model of wafer is equipment design dependent.

7.5.3.1 *WJOBJ State Model per Wafer Type* — Normally, state changes of Wafer Job are different per WaferType such as product wafer, non-product wafer (dummy or monitor), and jig wafer. To support those differences, equipment may have different WJOBJ State Models per WaferType.

7.5.4 *Services and Events of WJOBJ* — Services and events of WJOBJ are represented by WFJOBJ for seamless access to single or a group of WJs.

7.5.5 *Introduction of Event Representation Scheme by WJOBJ* — This Standard provides 'Event Representation' scheme by WJOBJ. WJOBJ can report events from other Standards as events from WJOBJ through WFJOBJ to correlate to ObjID of the WJOBJ for unified operation and better index capabilities. Events to be represented are equipment defined.

7.6 *Introduction of WFJ* — In order to provide a capability to manage multiple WJs in the equipment from the host, this Standard introduces WFJ which manages multiple WJs.

7.6.1 *Flow Management of WJs* — WFJ manages the flow of WJs as WJFlow. The order of WJ for product wafer may be determined by PJ. WFJ may manage multiple flows for different interests of order. For example:

- ProductionFlow as an ordered list of WJs created by PJs waiting for process.
- DummyFlow as an ordered list of WJs for dummy wafers waiting for its use.
- ExecutionFlow as an ordered list of WJs from both ProductionFlow and DummyFlow in actual process order at process module.
- CompletionFlow as an ordered list of WJs created by PJs and completed its process.

Note that methods to manipulate (such as add, delete, reorder, or lock) WJFlow may be added in future.

7.6.2 *Group Management of WJs* — In order to manage multiple WJs in the equipment from the host efficiently, WFJ provides a capability to deal multiple WJs as a group.

7.6.2.1 *Process Job Group (PJG)* — PJG consists of WJs which are created by the same parent PJ. For example, by using PJG for WJ addressing, the host can set Variable Parameters per wafer to the group of WJs at once after PJ creation.

7.6.2.2 *Execution Group (EXG) (Optional)* — EXG consists of WJs which are dealt at a same time such as a process batch. EXG may be assigned to some states of WJ State Model in which a batch is formed. EXG may also be re-assigned to some other states of WJ State Model for different size of batch. By using this group for WJ addressing, the host can deal WJs in a batch at a time with single service message. WFJ is responsible to assign PRJobID uniquely.

7.6.3 *NoPJInterrupt Capability* — WFJ provides a mode which prohibits to insert any other WJ in a WJ flow created by one PJ.

7.7 *Introduction of WFJOBJ* — In order to access to a WFJ in the equipment from the host, this Standard introduces WFJOBJ as a representation of a WFJ.

SUPERSEDED

Documents Document Sharing Hub
Full document details: SEMI E174-1116