No.	*Current State*	*Trigger*	*New State*	*Action*	*Comment*
11	EXECUTING	Processing of the last wafer of a current lot completed. The prober has received START command for next lot.	SETTING UP	None	None
12	STOPPING	The prober completes clearing of data and no alarm remains.	IDLE	None	None
13	PAUSING	The prober stopped the processing of the current wafer which was in EXECUTING state at a possible stop breakpoint and attained a safe state.	PAUSED	The prober is waiting for command (RESUME, STOP or ABORT).	None
14	ALARM PAUSED	Alarm is cleared.	PAUSED	The prober is waiting for command (RESUME, STOP or ABORT).	None
15	PROCESS PAUSE	An alarm occurs in the prober.	ALARM PAUSED	The prober is waiting for all alarms to be cleared, STOP or ABORT command.	None
16	PAUSED	RESUME command was received.	CHECKING	Verification of process program parameters.	Before continuing processing, the host or operator is required to issue RESUME command.
17	PAUSED	The operator instructed a state transition.	PAUSED SETTING UP	Probe card replacement or inker replacement is carried out by operator's instruction.	None
18	PAUSED SETTING UP	The operator instructed a state transition.	PAUSED	None	None
19	PAUSE	The prober received STOP command.	STOPPING	The prober starts clearing data.	Data is saved and valid.
20	PAUSE	The prober received ABORT command.	ABORTING	If possible, unsafe condition is reset.	Wafer data or lot data may be invalid or not available.
21	STOPPING	The prober received ABORT command.	ABORTING	If possible, unsafe condition is reset.	Wafer data or lot data may be invalid or not available.
22	ABORTING	An abort performance has completed.	IDLE	None	Only a state transition to IDLE is permitted.
23	IDLE	An alarm occurs.	IDLE with ALARMS	The prober is waiting for all alarms to be cleared.	None
24	IDLE with ALARMS	All alarms were cleared.	IDLE	None	The IDLE state is free of alarms.
25	IDLE	The operator instructed a state transition.	MAINTEN-ANCE	None	Maintenance and inspection of the prober modules and creation, change and deletion of process programs are possible.
26	MAINTEN-ANCE	The operator instructed a state transition.	IDLE	None	None

5.2.5 *The Process Model Condition Table*

5.2.5.1 Undermentioned Table 2 is an explanation of the transition condition of transition No. 10 in PSEM Processing State Model diagram.

Table 2 The Process Model Condition Table

Conditions	*Next State*
The check approves that the process program is changed.	SETTING UP
The check approves that the process program is **not** changed.	
The previous state was SETTING UP.	SETTING UP
The previous state was EXECUTING.	EXECUTING

5.3 *Prober Job State Model*

5.3.1 *Purpose*

5.3.1.1 The Prober Job state model is defined for the purpose of keeping track of its processing state of the material to be processed by the prober in every material. The host can give instructions on how to process which material by creating Prober Job to the equipment. Prober Job is created by the JOB CREATE command. Prober Job state exists in each Prober Job. It occurs upon creation of Prober Job and is deleted upon its completion. The host can control progress of material processing which is requested to the equipment according to an event report accompanied by progress of Prober Job.

5.3.2 *Prober Job State Model Diagram*

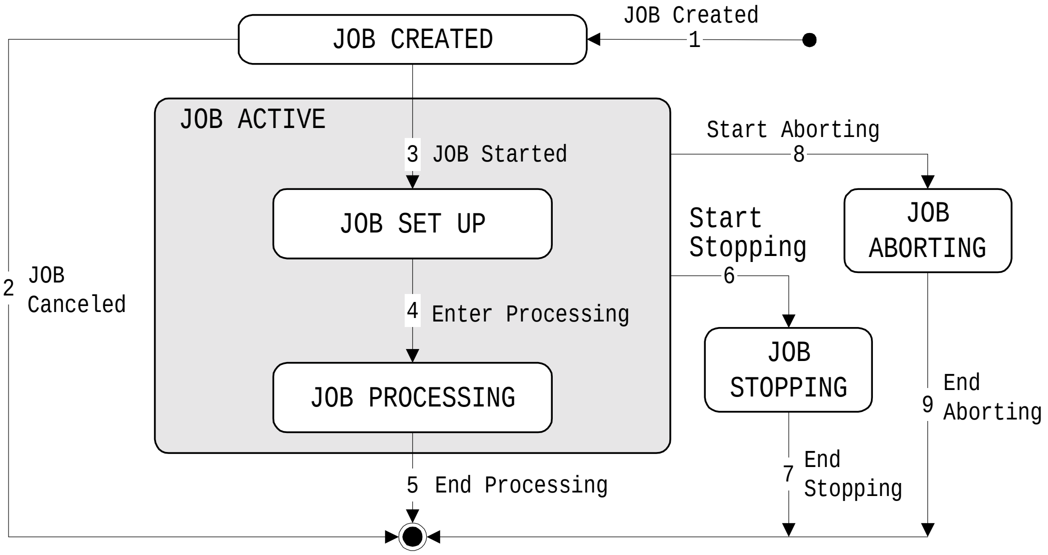

Figure 3
Prober Job State Model Diagram

5.3.3 *Description of Prober Job State*

5.3.3.1 *JOB CREATED* — State where Prober Job is created to the equipment Prober Job has not processed yet. It has nothing to do with whether material reached the equipment or not.

5.3.3.2 *JOB ACTIVE* — State where normal Prober Job processing operation is performed It consists of two sub-states.

5.3.3.3 *JOB SET UP (JOB ACTIVE Sub-state)* — State that waits for condition to be ready for processing such as Equipment preparation.

5.3.3.4 *JOB PROCESSING (JOB ACTIVE Sub-state)* — State where this processing is performed.

5.3.3.5 *JOB STOPPING* — State where attempts are made to stop Prober Job.

5.3.3.6 *JOB ABORTING* — State where attempts are made to abort Prober Job.

5.3.4 *Prober Job State Transition Table*

Table 3 Prober Job State Transition Table

No.	*Current State*	*Trigger*	*New State*	*Action*	*Comments*
1	Undefined	JOB_CREATE command was received from host or the Job Create operation was instructed by operator when Processing State is neither INIT nor MAINTENANCE state.	JOB CREATED	Create Prober Job.	None.
2	JOB CREATED	JOB_CANCEL command from host or operator.	Undefined	Cancel Prober Job.	None.
3	JOB CREATED	START command from host or the Start operation was instructed by operator.	JOB SET UP	Prepare for processing of Prober Job and wait for processing to be possible.	None.
4	JOB SET UP	Prober Job processing conditions were established.	JOB PROCESSING	Start processing of the main unit.	None.
5	JOB PROCESSING	Processing of Prober Job was normally completed.	Undefined	Delete Prober Job.	None.
6	JOB ACTIVE	STOP command was received from host or the stop operation was instructed by operator.	JOB STOPPING	Start stop processing.	A trigger may be caused by error detection of the equipment itself as well as by instruction from outside.
7	JOB STOPPING	The stop operation was completed.	Undefined	Delete Prober Job.	None.
8	JOB ACTIVE	ABORT command was received from host of the Abort operation was instructed by operator.	JOB ABORTING	Start abort processing.	A trigger may be caused by error detection of the equipment itself as well as by instruction from outside.
9	JOB ABORTING	Abort processing was completed.	Undefined	Delete Prober Job.	None.

6 Collection Event List

6.1 *Requirements*

6.1.1 All GEM-required Events are required by the PSEM. Since Processing State Model is required by the PSEM, all state transitions are required Events.

6.1.2 All GEM-required Events associated with the GEM Control, Communications, Alarm, and Spooling State Models are required. This section of the PSEM lists only those collection events that are not associated with a change of state or those requiring specific data variables (DVVALs) or Reports defined in the PSEM.

6.2 *Collection Event Tables*

6.2.1 Table 4 shows events and reports required for the PSEM process state transition. Table 5 shows events and reports required for the Prober Job state transition. Table 6 shows other collection events.

Table 4 Collection Events Required by Process State Transition

Transition	*Transition No.*	*Current State*	*New State*	*Typical Variable Data*
Start INIT	1	Undefined	INIT	Previous Process State
Into IDLE	2	INIT	IDLE	
	5	EXECUTING		
	12	STOPPING		
	22	ABORTING		
	24	IDLE with ALARMS		
	26	MAINTENANCE		
Into IDLE with ALARMS	23	IDLE	IDLE with ALARMS	
Into MAINTENANCE	25	IDLE	MAINTENANCE	
Start SETTING UP	3	IDLE	SETTING UP	
	11	EXECUTING		
	10	CHECKING		
Start EXECUTING	4	SETTING UP	EXECUTING	
	10	CHECKING		
Start PAUSING	9	SETTING UP	PAUSING	
	9	EXECUTING		
Into PAUSED	13	PAUSING	PAUSED	
	18	PAUSED SETTING UP		
	14	ALARM PAUSED		
Start CHECKING	16	PAUSED	CHECKING	
Into PAUSED SETTING UP	17	PAUSED	PAUSED SETTING UP	
Into ALARM PAUSED	8	SETTING UP	ALARM PAUSED	
	8	EXECUTING		
	15	CHECKING		
	15	PAUSING		
	15	PAUSED SETTING UP		
	15	PAUSED		
Start STOPPING	6	SETTING UP	STOPPING	
	6	EXECUTING		
	19	CHECKING		
	19	PAUSING		
	19	PAUSED SETTING UP		
	19	PAUSED		
	19	ALARM PAUSED		
Start ABORTING	7	SETTING UP	ABORTING	
	7	EXECUTING		
	20	CHECKING		
	20	PAUSING		
	20	PAUSED SETTING UP		
	20	PAUSED		
	20	ALARM PAUSED		
	21	STOPPING		

Table 5 Collection Events Required by Prober Job State Transition

Transition	*Current State*	*New State*	*Typical Variable Data*
JOB Created	Undefined	JOB CREATED	EventJobID
JOB Canceled	JOB CREATED	Undefined	
JOB Started	JOB CREATED	JOB SET UP	
Enter Processing	JOB SET UP	JOB PROCESSING	
End Processing	JOB PROCESSING	Undefined	
Start Aborting	JOB SET UP, JOB PROCESSING	JOB ABORTING	
End Aborting	JOB ABORTING	Undefined	
Start Stopping	JOB SET UP, JOB PROCESSING	JOB STOPPING	
End Stopping	JOB STOPPING	Undefined	

Table 6 Other Collection Events

Event Name	*Contents*	*Typical Variable Data*
Wafer Start	Wafer processing was started.	WaferStartJobID, WaferStartWaferID
Wafer End	Wafer processing was completed.	WaferEndJobID, WaferEndWaferID
Ready to Receive Previous Data	Waits for reception of instructed data for wafer.	WaitPreDataJobID, WaitPreDataWaferID

7 Data Item Variables

7.1 The purpose of this section is to define the list of data item variables required by the PSEM. Values of these variables will be available to the host through collection event reports and host status queries.

7.2 *Requirements*

7.2.1 All generic variable data items defined in GEM are required by all PSEM equipment. Any supplier-defined variables shall be documented in the same format used by this document. The following minimum information is required:

<variable name> Class: <ECV, SV, or DVVAL>

Format: <SML>

Description: <if class=DVVAL, description must contain statement of when data is valid>.

<If format = ASCII then a length is required. It is assumed to be left-justified unless otherwise noted. >

7.3 *Data Types*

7.3.1 Equipment Constants (ECVs) can be changed by the host using S2F15. The operator may be able to change some values, but the equipment does not change the values on its own. The value of an equipment constant may be queried by the host at any time, using the S2F13/14 transaction. They reside in non-volatile memory of the equipment. Equipment constants remain in effect until they are overwritten either by manual entry or by a NEW EQUIPMENT CONSTANT SEND. Equipment constants have various uses in PSEM, including the following:

- Equipment offsets that match the performance of several pieces of equipment that would otherwise perform differently due to inherent manufacturing differences. Examples are home values and motion axis scaling factors.
- Setting the configuration of the equipment to allow for different material specifications, equipment options, material flows, frequency of automatic functions, etc. Examples are yield check frequency.
- Managing optional machine features. Examples are constants that tell the system whether optional features such as automated media stackers are present and control the configuration and function of these optional subsystems when they are present.

7.3.2 Status Variables (SVs) are valid at all times. A SV may not be changed by the host but may be changed by the equipment or operator. The value of status variables may be queried by the host at anytime using the S1,F3/4 or S6,F19/20 transactions. DVVALs are variables that are valid only upon the occurrence of specific collection events. An attempt to read a data variable at the wrong time will not generate an error, but the data reported may not have relevant meaning.

7.4 *Data Item Variable Table*

Table 7 Data Item Variable Table

Variable Name	*Description*	*Class*	*Format*	*Comments*
StopUnit	Constant which specifies separation of stop operation in probing.	ECV	U1	0=Unit of Die 1=Unit of Wafer 2=Unit of Cassette 3=Unit of Lot
EventJobID	Prober Job ID which occurs Prober Job state transition.	DVVAL	A[30]	Valid when Prober Job state transition occurs.
EventJobState	Current state of Prober Job whose state changed.	DVVAL	U2	Valid in event of Prober Job state transition.
ResultData	Processing result in unit of one wafer.	DVVAL	L[n]	Valid only in wafer end event.

8 Process Program Management

8.1 PSEM explains how to specify a process program for the prober and the process program control between the host and prober.

8.2 *Process Program Requirements*

8.2.1 The PSEM requires that the GEM capability of process program management be fully supported for this class of prober.

8.3 *Process Program*

8.3.1 The process program of the prober consists of the main process program and two or more sub process programs that are referred to by the main process program.

Table 8 Kinds and Descriptions of Process Programs

Kind	*Description*
Product type master record	Master process program for processing.
Sub-parameter record	Sub-process program for processing. Various kinds of parameter records that consist main process program for processing.

8.3.2 These process programs are controlled by the prober with grouped into different categories according to application purposes and structures. The prober defines PPID with “structure class name/class name/process program record name.” The number of PPID digits is defined by an equipment supplier. If the number of the digits is less than a default value, left-justify and fill in space after PPID. Do not use space at some midpoint in PPID. At present, the prober practically uses the structure class name and class name as shown in the table given below.

Table 9 How to Operate Class Name and Structural Class Name

Classification	*Operation*	*Remarks*
Structural class name	Class name indicating major classification of process program record.	Defined by supplier.
Class name	Class name indicating attribute of process program record.	Defined by supplier.

8.4 *Master Process Program Record*

8.4.1 The master process program record is a plain text (ASCII) record like below:

DEVICE:xxxxxxxx

WAFER_SIZE:xx

DIE_SIZE:xxxxxx,xxxxxx

....

PARAM_RECORD00:structure class name/class name/process program record name

....

PARAM_RECORDnn:structure class name/class name/process program record name

8.4.1.1 Each line has a structure:
TAG + ":" + parameter(s) + line terminator

8.4.1.2 The line terminator is some of CR+LF (0x0d0a), CR (0x0d), LF (0x0a).

8.4.1.3 Specific parameters of equipment supplier may be put between "HOT_CHUCK_TOLE" and "PARAM_FILE00".

Table 10 Tag Parameters in Master Process Program

Tag	*Description*	*Format*	*Unit*
PRODID	Product name (i.e., Device type)	A[1,24]	-
WAFER_SIZE	Wafer size	A[1,3]	mm
DIE_SIZE_X	Die size X	A[1,6]	micron
DIE_SIZE_Y	Die size Y	A[1,6]	micron
MULTI_DIE	Multi probing number	A[1,3]	-
REFERENCE_DIE_COORD_X	Reference die coordinator X	A[1,4]	-
REFERENCE_DIE_COORD_Y	Reference die coordinator Y	A[1,4]	-
OVER_DRIVE	Over drive value	A[1,4]	micron
FLAT_ANGLE	Flat orientation	A[1,3]	degree
FLAT_TYPE	FLAT or NOTCH	A[1,5]	-
HOT_CHUCK	Flag of hot chuck usage	A[2,3] “ON” or “OFF”	-
HOT_CHUCK_TEMP	Hot chuck temperature	A[1,4]	Celsius
HOT_CHUCK_TOLE	Hot chuck tolerance	A[1,3]	Celsius
PARAM_RECORDnn	Sub-parameter record nn	A[ ]	-

9 Map Data

9.1 This section defines handling of map data in the PSEM equipment. The contents of map data are not defined here. The specification in this section is provisional. This will be revised in the near future.

9.2 *Use of Map Data*

9.2.1 Map data is divided into two groups: result data which is reported to the host when processing of each wafer is completed and instruction data which is instructed by the host in a unit of wafer at the time of re-inspection or marking.

9.3 *Structure of Map Data*

9.3.1 The map data has a list structure and the result data is defined by class <DVVAL> and the instruction data is defined by <CEPVAL>.

9.3.1.1 Example:

```
<ResultData>
    L,n
      1. <Data1>
      :
      n. <Data_n>
```

9.4 *Uploading (Reporting) Result Data*

9.4.1 Result Data is reported by S6,F11 upon completion of the inspection in each wafer.

9.4.1.1 Example:

```
L,3
  1. <DATAID>
  2. <CEID>:Wafer End
  3. L,a
      1. L,2
          1. <RPTID1>
          2. L,7
              1. <LOTID>
              2. <PROCID>
              3. <IDTYP>
              4. <MID>
              5. <ROWCOLCT>
              6. <REFDIEPOS>
              7. <ResultData>
      :
      a. L,2
          1. <RPTIDa>
          2. L,b
              1. <LOC>
              2. <PRODID>
              3. <WAFSIZE>
              4. <FLATTYPE>
              5. <FLATANGL>
              6. <REFDIECOORD>
              7. <BINLIST>
              :
              b. <Vb>
```

9.5 *Downloading Instruction Data*

9.5.1 If the past inspection result is required in processing a wafer, the equipment obtains instructed data from the host in the following procedure.

Comment	*Host*		*Equipment*	*Comment*
				Transfer a wafer. The equipment is ready to receive previous result data for a wafer concerned.
		←	S6,F11	"Waiting Previous Data (wait for reception of previous result data)" event report.
Check event.	S6,F12	→		
Instruction data download (PRE-DATA_DOWNLOAD)	S2,F49	→		
		←	S2,F50	Reception response. The processing is performed based on instruction data.

9.5.2 An example of Waiting Previous Data event report S6,F11 in the procedure above is shown below:

9.5.2.1 Example:

```
S6,F11
     L,3
        1. <DATAID>
        2. <CEID>     :Waiting Previous Data
        3. L,2
              1. <RPTID1>
              2. L,4
                    1. <LOTID>   :Prober Job ID
                    2. <PROCID>
                    3. <MID>
                    4. <IDTYP>
```

9.5.3 An example of remote command S2,F49 used in the procedure above is shown below.

9.5.3.1 Example:

```
S2,F49
     L,4
        1. <DATAID>
        2. <OBJSPEC>  :a null length item.
        3. < RCMD>     :"PRE-DATA_DOWNLOAD"
        2. L,n
              1. L,2
                    1. <CPNAME1>
                    2. <CEPVAL1>
              :
              n. L,2
                    1. <CPNAMEn>    :PreviousResultData
                    2. L,b
                          1. <CPVALn1>
                          :
                          b. <CPVALnb>
```

9.6 *Map Data Item Variable Table*

Table 11 Map Data Item Variable Table

Variable Name	*Description*	*Format*	*Comments*
LOTID	Lot ID (Prober Job ID)	A[30]	
PROCID	Process ID	A[20]	
LOC	Cassette location	U1	
PRODID	Product name (i.e., Device type)	A[24]	
SLOTNO	Slot position in the cassette	A[2]	MID
WAFERNO	Wafer number	A[2]	MID
WAFERID	Wafer ID	A[32]	MID
IDTYP	ID type to recognize a wafer.	A[8]	"SLOTNO" or "WAFERNO" or "WAFERID"
WAFSIZE	Wafer size	A[3]	Unit is mm
FLAT	Flat or Notch	A[5]	"FLAT" or "NOTCH"
FLATANGLE	Flat angle	A[3]	Range from "000" to "359". Unit is degree.
ROW	Row count	U2	
COLUMN	Column count	U2	

DIESIZE_X	Die Size X	A[1,6]	Unit is micron
DIESIZE_Y	Die Size X	A[1,6]	Unit is micron
REFDIECOORD_X	X coordinator of reference die	A[1,4]	
REFDIECOORD_Y	Y coordinator of reference die	A[1,4]	
REFDIEPOS_X	X address of reference die from wafer center	I4	Unit is micron
REFDIEPOS_Y	Y address of reference die from wafer center	I4	Unit is micron
BINLIST	Bin value array containing pass/fail information.	B[n]	
ResultData	Result data	B[m]	

NOTE 1: One of these three OPT items needs to be specified.

9.6.1 The data structure of ResultData is selectable with soft-switch as below:

(i) X,Y,BIN,...,X,Y,BIN

(ii) X,Y,N,BIN,BIN,...,BIN

(iii) BIN,BIN,...,BIN

9.6.2 The BIN data is a binary array and defined by an equipment supplier. For example, BIN data may include an attribute of die (skip, on-wafer, etc.), pass or fail information and bin code.

9.7 *Data Item Variable Table*

Table12 Data Item Variable Table

Variable Name	*Description*	*Class*	*Format*	*Comments*
BinType	Constant which specifies the kind of the data structure of ResultData.	ECV	U1	0= X, Y, BIN, ..., X, Y, BIN 1= X, Y, N, BIN, BIN, ..., BIN 2= BIN, BIN, ..., BIN

10 Remote Commands

10.1 The purpose of this section is to identify remote commands, command parameters, and valid commands versus states in the processing state models. The specification in this section is provisional. This will be revised in the near future.

10.2 *Requirements*

10.2.1 The prober must support the GEM-required remote commands. (Some of the GEM required remote commands are restated here to define PSEM specific requirements.)

10.2.2 All the remote commands defined by PSEM are required.

10.2.3 The alphanumeric strings defined by PSEM for remote commands (RMCD) and command parameter (CPNAME) are required.

10.2.4 If additional remote commands are supported, then the “Remote Command Versus Valid States” matrix must be generated for these additional commands. Place an “X” in the table for each state in which a given command is valid.

10.3 *Remote Commands Descriptions*

10.3.1 *ABORT* — This command terminates the current processing. ABORT makes no guarantee about completion of the current process.

10.3.2 *JOB_CANCEL* — Instruct to delete Prober Job before starting processing.

10.3.3 *JOB_CREATE* — Instruct to create Prober Job in batch processing unit (usually, unit of one cassette). This allows management data for batch processing unit to be instructed to the prober.

10.3.4 *PRE-DATA_DOWNLOAD* — Downloads previous result data which instructs the details of processing for a wafer.

10.3.5 *ONLINE-LOCAL* — Change the control state from On-Line Remote to On-Line Local.

10.3.6 *ONLINE-REMOTE* — Change the control state from On-Line Local to On-Line Remote.

10.3.7 *PAUSE* — Instructs the equipment to pause the processing. This command puts the prober into PAUSING state when the current operation is completed.

10.3.8 *PP-SELECT* — This command instructs the prober to copy the indicated Process Program from non-volatile storage to the prober's Process Program execution area.

10.3.9 *RESUME* — Instructs the equipment in pause to resume operation. This command makes the equipment resume the processing from a point where the processing PAUSED.

10.3.10 *START* — Instructs to start Prober Job.

10.3.11 *STOP* — Instructs the equipment to stop the processing. This command completes the current processing unit, stops the equipment in safe state, and puts the equipment back to IDLE state. STOP has an intention of causing a normal completion after the current unit is completed. Also, whether the processing unit is die or wafer can be specified by the equipment constant Stop Unit.

10.4 *Parameter Related to Remote Command*

Table 13 Description of Remote Command

Command Name	*Command Parameter*			
	Name	*OPT/REQ*	*Description*	*Format*
ABORT			No parameter.	
JOB_CANCEL	ProberJobID	REQ.	ID of deleted Prober Job	A[30]
JOB_CREATE	ProberJobID	REQ.	ID of created Prober Job (Used as lot ID.)	A[30]
	LOC	REQ.	Location number of the cassette in which processed wafers are placed.	B
	PRODID	OPT	Product name i.e., Device type	A[24]
	PPID	OPT	Product type master record name. Unless otherwise specified, use the current PPID.	A[]
	NO-OF-WAFER	OPT	Number of wafers held in the cassette (regardless of whether the wafers are subject to processing or not).	A[20]
	SLOT-ORD	OPT	Specify the wafer pick-up order from the cassette.	BOOL
	SLOT-INFO	OPT	Set a wafer ID and whether a wafer is subject to processing or not. Example: L[25,26] 1. L[2] 1. A[1,28] ; wafer ID 2. B ; whether a wafer is subject to processing or not. : n. L[2] 1. A[1,28] ; wafer ID 2. B ; whether a wafer is subject to processing or not. (Notes) n = 25 or 26	L[25,26]

Command Name	*Command Parameter*			
	Name	*OPT/REQ*	*Description*	*Format*
PRE-DATA_DOWNLOAD	ProberJobID	REQ.	Prober Job ID	A[30]
	PROCID	REQ.	Process ID	A[20]
	LOC	OPT	Port number of the cassette in which wafers to be processed are placed.	B
	PRODID	OPT	Product name (i.e., Device type)	A[24]
	SLOTNO	OPT (See NOTE 1.)	Slot number in which the wafer is held.	A[2]
	WAFERNO	OPT (See NOTE 1.)	Wafer number set by the host or operator.	A[2]
	WAFERID	OPT (See NOTE 1.)	Wafer ID read by OCR.	A[28]
	IDTYP	REQ.	ID type to recognize a wafer.	A[8]
	WAFSIZE	OPT	Wafer size	A[3]
	FLAT	OPT	Flat or Notch	A[5]
	FLATANGLE	OPT	Flat angle	A[3]
	ROW	REQ.	Row count	U2
	COLUMN	REQ.	Column count	U2
	DIESIZE_X	OPT	X die size	A[6]
	DIESIZE_Y	OPT	Y die size	A[6]
	REFDIECOORD_X	REQ.	X coordinator of reference die	A[4]
	REFDIECOORD_Y	REQ.	Y coordinator of reference die	A[4]
	REFDIEPOS_X	REQ.	X address of reference die from wafer center	I4
	REFDIEPOS_Y	REQ.	Y address of reference die from wafer center	I4
	BINLIST	OPT	Bin value array containing pass/fail information.	B[n]
	PreviousResultData	REQ.	Previous result data information	L[n]
ONLINE-LOCAL			No parameter	
ONLINE-REMOTE			No parameter	
PAUSE			No parameter	
PP-SELECT	PPID	REQ.	Product type master record name.	A[]
RESUME	Resume-Die	OPT	Specify which die to restart the processing (e.g., last die, current die, next die).	B
START	ProberJobID	REQ.	ID of Prober Job to be started.	A[30]
STOP			No parameter	

NOTE 1: At least one of <SLOTNO>, <WAFERNO>, and <WAFERID> need to be set.

10.5 *Remote Commands and PSEM Process Model Mapping*

10.5.1 Each remote command either can or cannot be executed depending on a state other than the control state.

10.5.2 The following table shows remote commands and the relationship between process state to be operated by these commands and Prober Job state.

Table 14 Remote Command vs. Process State/Prober Job State

Command Name											
	STOP	START	RESUME	PP-SELECT	PAUSE	ONLINE-REMOTE	ONLINE-LOCAL	PRE-DATA_DOWNLOAD (See NOTE 3.)	JOB_CREATE	JOB_CANCEL	ABORT
PROCESSING STATE											
INIT											
IDLE				X (See NOTE 1.)		X (See NOTE 2.)	X (See NOTE 2.)				
IDLE with ALARMS											
MAINTENANCE											
PROCESSING ACTIVE											
...PROCESS											
......SETTING UP	X				X						X
......EXECUTING	X				X						X
...PAUSE											
......PROCESS PAUSE											
.........PAUSING	X										X
.........PAUSED	X		X								X
.........CHECKING	X										X
.........PAUSED SETTING UP	X										X
......ALARM PAUSED	X										X
...STOPPING											X
...ABORTING											
Prober Job STATE											
Undefined				X (See NOTE 1.)		X (See NOTE 2.)	X (See NOTE 2.)		X		
JOB CREATED		X								X	
JOB ACTIVE											
...JOB SET UP								X (See NOTE 3.)			
...JOB PROCESSING								X (See NOTE 3.)			(X)
JOB STOPPING											
JOB ABORTING											

NOTE 1: Both IDLE and Undefined need hold true for PP-SELECT.

NOTE 2: Both IDLE and Undefined need hold true for ONLINE-REMOTE and ONLINE-LOCAL.

NOTE 3: PRE-DATA_DOWNLOAD is valid when Ready to Receive Previous Data event is reported.

10.6 *Restriction on the Operator by Control States*

10.6.1 For the remote commands, the operator of the prober or the host is restricted by the control states of the equipment.

Table 15 Table of Restrictions on the Operator by Control States

Remote Commands	*Operator*		*Host*	
	LOCAL	*REMOTE*	*LOCAL*	*REMOTE*
JOB_CREATE	O	X	O	O
PP-SELECT	O	X	O	O
PRE-DATA _DOWNLOAD	X	X	O	O
JOB_CANCEL	O	X	O	O
START	O	X	X	O
PAUSE	O	O	X	O
RESUME	O	O	X	O
STOP	O	O	X	O
ABORT	O	O	X	O
ONLINE-LOCAL	X	O	X	O
ONLINE-REMOTE	O	X	O	X

O: Operation is allowed.

X: Operation is prohibited.

11 Scenario

11.1 The purpose of this section is to document a typical PSEM specified operation scenario.

11.2 *From Power-On to Online Remote*

11.2.1 A typical scenario of transition from equipment power-on to online remote is shown below.

Comment	*Host*		*Equipment*	*Comment*
				(Equipment power-on)
				(Equipment initialization operation)
		←	S1,F13	Communication establishment request
Communication establishment request confirmation	S1,F14	→		
				(Press the online transition switch.)
		←	S1,F1	Are You There Request
Online data	S1,F2	→		
				(Online transition is completed.)
		←	S6,F11	Send ONLINE-REMOTE transition event report
Confirm event report	S6,F12	→		
		←	S6,F11	Send IDLE transition event report
Confirm event report S6,F12	S6,F12	→		

11.3 *1 Lot Processing in Online Remote*

11.3.1 A typical scenario from 1 lot processing instruction to processing completion in online remote is shown below.

Comment	*Host*		*Equipment*	*Comment*
				ONLINE-REMOTE IDLE
Send JOB_CREATE command	S2,F49	→		
		←	S2,F50	Confirm host command
		←	S6,F11	Send Prober Job Created event report
Confirm event report	S6,F12	→		
				(Carry in cassette)
		←	S6,F11	Send material carry-in event report
Confirm event report	S6,F12	→		
Send START command	S2,F49	→		
		←	S2,F50	Confirm host command
		←	S6,F11	Send Prober Job Started event report
Confirm event report	S6,F12	→		
		←	S6,F11	Send SETTING UP transition event report
Confirm event report	S6,F12	→		
				(Equipment setup is started.)
				(Equipment setup is completed.)
		←	S6,F11	Send Enter Processing event report
Confirm event report	S6,F12	→		
		←	S6,F11	Send EXECUTING transition event report
Confirm event report	S6,F12	→		
				[DO] "wafer count times"
				(Wafer load is completed.)
				[IF]
				Is Instruction data required?
				[THEN]
		←	S6,F11	Send Waiting Previous Data event report
Confirm event report	S6,F12	→		
Send PRE-DATA_DOWNLOAD command	S2,F49	→		
		←	S2,F50	Confirm host command
				[END_IF]
				(Wafer processing is started.)
				(Wafer processing is completed.)
		←	S6,F11	Send wafer end event report (Result Data)
Confirm event report	S6,F12	→		
				(Wafer unload is completed.)
				[END_DO]
		←	S6,F11	Send End Processing event report
Confirm event report	S6,F12	→		
		←	S6,F11	Send IDLE transition event report
Confirm event report	S6,F12	→		
				(Carry out cassette)
		←	S6,F11	Send material carry-out event report
Confirm event report	S6,F12	→		

NOTE 1: Each event report send order in Prober Job Started and SETTING UP transition, Enter Processing and EXECUTING transition, and End Processing and IDLE transition may be inverted.

12 GEM Addition Request

12.1 The purpose of this section is to specify all GEM addition requests which are required to support the equipment of this class.

12.2 *Requirements*

12.2.1 GEM addition requests required of PSEM are as shown below:

- Establish Communications,
- Dynamic Event Report Configuration,
- Variable Data Collection,
- Status Data Collection,
- Alarm Management,
- Remote Control,
- Equipment Constants,
- Process Program Management,
- Equipment Terminal Services,
- Clock,
- Spooling, and
- Control (host-initiated).

NOTICE: SEMI makes no warranties or representations as to the suitability of the standard set forth herein for any particular application. The determination of the suitability of the standard is solely the responsibility of the user. Users are cautioned to refer to manufacturer's instructions, product labels, product data sheets, and other relevant literature respecting any materials mentioned herein. These standards are subject to change without notice.

The user's attention is called to the possibility that compliance with this standard may require use of copyrighted material or of an invention covered by patent rights. By publication of this standard, SEMI takes no position respecting the validity of any patent rights or copyrights asserted in connection with any item mentioned in this standard. Users of this standard are expressly advised that determination of any such patent rights or copyrights, and the risk of infringement of such rights, are entirely their own responsibility.

Copyright by SEMI® (Semiconductor Equipment and Materials International), 3081 Zanker Road, San Jose, CA 95134. Reproduction of the contents in whole or in part is forbidden without express written consent of SEMI.

SEMI E93-0200 (Withdrawn 0703)
PROVISIONAL SPECIFICATION FOR CIM FRAMEWORK ADVANCED PROCESS CONTROL COMPONENT

This provisional specification was technically approved by the Global Information & Control Committee and is the direct responsibility of the North American Information & Control Committee. Current edition approved by the North American Regional Standards Committee on April 11, 2003. Initially available at www.semi.org June 2003; to be published July 2003. Originally published September 1999, previously published February 2000.

NOTICE: This document was balloted and approved for withdrawal in 2003.

1 Purpose

1.1 The Advanced Process Control (APC) component supports the Process Machine component' s execution of process machine jobs by optimizing machine-specific settings for the current material and machine state to achieve desired process effects. It may also detect faults in processing and recommend an appropriate response to the Process Machine component. An APC component may include statistical process control, model-based process control, multi-variate analysis, trace analysis, fault pattern matching, or other analysis techniques.

1.2 The Advanced Process Control component specification does not constrain implementation approaches. However, subcomponents are defined to enable integration with different types of analysis and computation mechanisms, specifically those provided in other software systems. The subcomponents are defined principally along boundaries expected to align with products addressing process control in the factory. The number of subcomponents offered by suppliers may vary, with some suppliers supporting all of the subcomponent interfaces and others focusing on a specific subset of subcomponents. These subcomponent boundaries provide the customer with added flexibility to integrate and use multiple suppliers’ products.

1.3 This specification provides the interfaces required by Manufacturing Execution Systems to:

- Define control data structures to support control algorithms.
- Launch, coordinate, and monitor the execution of control processing and analysis algorithms.

Copyright by SEMI® (Semiconductor Equipment and Materials International), 3081 Zanker Road, San Jose, CA 95134. Reproduction of the contents in whole or in part is forbidden without express written consent of SEMI.

SEMI E94-0705
SPECIFICATION FOR CONTROL JOB MANAGEMENT

This specification was technically approved by the global Information & Control Committee. This edition was approved for publication by the global Audits and Reviews Subcommittee on April 7, 2005. It was available at www.semi.org in June 2005 and on CD-ROM in July 2005. Originally published February 2000; previously published November 2004.

1 Purpose

1.1 This specification describes equipment provided services to the factory that supports a high level of factory automation. These services provide capabilities for the host to coordinate processing and disposition of materials on production equipment.

2 Scope

2.1 This specification may be applied to equipment that is compliant to SEMI E30 (GEM). However, it is also intended that this standard will be useful for future generation equipment interfaces that supercede SEMI E30, such as SEMI E53.

NOTICE: This standard does not purport to address safety issues, if any, associated with its use. It is the responsibility of the users of this standard to establish appropriate safety and health practices and determine the applicability of regulatory or other limitations prior to use.

3 Limitations

3.1 This standard presents a model of the ControlJob. The model delineates the services (messages) and behavior of the ControlJob. The model is protocol independent. Thus, an ancillary standard must be selected in order to provide a complete implementation.

3.2 This standard should not be applied to non-production equipment, such as, material transport systems or facilities (environmental) controllers.

3.3 This specification applies to equipment for which the atomic unit of material is the same for all input and output carriers on the equipment. It may not apply to equipment which would perform operations such as slicing or assembly that would require or result in different input and output material objects. This specification may not apply to equipment or equipment configurations where the equipment does not handle carriers, as in the case of a stepper or scanner in a linked photolithography cell.

4 Referenced Standards and Documents

4.1 *SEMI Standards*

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

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

SEMI E40 — Standard for Processing Management

SEMI E53 — Event Reporting

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

5 Terminology

5.1 *Definitions*

5.1.1 *control job* — defines a unit of work on equipment for one or more carriers. The work is described by a set of one or more process jobs to be applied to the material contained in the carriers.

5.1.2 *de-queue* — the act of removing an item from a queue. The de-queue implies nothing about the status of the item after removal.

5.1.3 *equipment* — the intelligent system that communicates with the host.

5.1.4 *host* — the intelligent system that communicates with the equipment.

5.1.5 *life cycle* — the processes and activities of something from its beginning (creation) to its ending.

5.1.6 *multi-module equipment* — equipment that has more than one distinct processing resource (e.g., chamber).

5.1.7 *production equipment* — equipment that measures or adds value to the product.

5.1.8 *protocol independent* — for software, this means that the message descriptions are independent of delivery mechanisms.

5.1.9 *set-up* — a description of the current process capability of an equipment.

5.1.10 *substrate* — basic unit of material on which work is performed to create a product. Examples include wafers, lead frames, CD's, die, flat panel displays, circuit boards, and disks.

5.1.11 *substrate port* — the carrier location from which substrates are accessed by the equipment.

5.1.12 *uni-carrier* — term for an equipment mode of operation in which all material is returned to the source carrier after processing.

5.1.13 *user start* — activities that are initiated on a system by another system or operator.

6 Conventions

6.1 *Object Models*

6.1.1 This standard uses object models to specify the control job interface.

6.1.2 *Object Services Standard*

6.1.2.1 This document conforms to the conventions established by SEMI E39.

6.1.3 *Formal Name of an Object*

6.1.3.1 The text capitalizes formal object name references, similar to the way capitalization is normally used when discussing entities. When describing something in the general (like cities) lower case is used, but when a specific entity is of interest (New York City), then first letters are capitalized.

6.2 *State Model Methodology*

6.2.1 This document uses the Harel state chart convention for describing dynamic operation of defined objects. The outline of this convention is described in an attachment of SEMI E30. The official definition of this convention is described in "State Charts: A Visual Formalism for Complex Systems" written by D. Harel in Science of Computer Programming 8, 1987.[1]

6.2.2 The Harel convention does not have the concept of state models of "creation" and "extinction" for expressing a temporary entity. The "job" described in this document is such an entity, and a copy of the same state model is used for an independent job newly created. In this document, a circle with a black circle inside is used for expressing extinction of an entity. A filled, black circle denotes the entry to the state model (the entity creation).

6.2.3 Transition tables are provided in conjunction with the state diagrams to explicitly describe the nature of each state transition. A 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:

1) Actions taken upon exit of the previous state.

2) Actions taken upon entry of the new state.

3) Actions taken which are most closely associated with the transition.

6.2.3.1 No differentiation is made between these cases.

[1] Elsevier Science, P.O. Box 945, New York, NY 10159-0945, http://www.elsevier.nl/homepage/browse.htt

6.2.4 The state models included in this standard are a requirement for Control Job Management compliance. A state model consists of a state model diagram, state definitions, and a state transition table. When using collection events, all state transitions in this standard, unless otherwise specified, shall correspond to collection events.

6.2.5 A state model represents the host's view of the equipment, and does not necessarily describe the internal equipment operation. When using collection events, all Control Job Management 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 Control Job Management state models for transition to another state. In the case, the equipment makes the required transition without any additional actions in this situation.

Num	*Previous State*	*Trigger*	*New State*	*Actions*	*Comments*

6.3 *Service Message Representation*

6.3.1 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.

6.3.2 *Service Definition*

6.3.2.1 A service definition table defines the specific set of messages for a given service resource, as shown in the following table:

Message Service Name	*Type*	*Description*

6.3.2.2 Type can be either "N" = Notification or "R" = Request & Response.

6.3.2.3 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. 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).

6.3.3 *Service Parameter Dictionary*

6.3.3.1 A service parameter dictionary table defines the description, format and its possible value for parameters used by services, as shown in the following table:

Parameter Name	*Description*	*Format: Possible Value*

6.3.3.2 A row is provided in the table for each parameter of a service.

6.3.4 *Service Message Definition*

6.3.4.1 A service message definition table defines the parameters used in a service, as shown in the following table:

Parameter	*Req/Ind*	*Res/Cnf*	*Comment*

6.3.4.2 The columns labeled REQ/IND and RSP/CNF 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" or the request. The receiver may then send a "Response" which the original sender terms the " Confirmation".

6.3.4.3 The following codes appear in the REQ/IND and RSP/CNF columns and are used in the definition of the parameters (e.g., how each parameter is used in each direction):

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 value of the other parameter.
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).

7 Overview

7.1 This section provides an overview of the control job functionality. It does not contain the specifications which define that functionality.

7.1.1 Control jobs provide a supervisory level of control for process jobs on material processing equipment. They can be used to reduce the amount of host level interaction required for material processing. A factory host is provided with methods for instructing the equipment to provide only significant factory level events, such as, a carrier complete. The ControlJob also supplies methods for the disposition of material after processing.

7.2 *User Requirements*

7.2.1 To handle the complexity required for manufacturing, equipment must support the ability to coordinate its processing services with the factory’s needs. The ControlJob provides the services that the factory needs to accomplish this coordination. The requirements that the ControlJob satisfies include: (1) a method by which the equipment coordinates related work, for instance, all process jobs associated with a carrier, and (2) a method by which the equipment can be informed of material destination after processing. The ControlJob is not a type of process job. It is not responsible for the coordination of the processing resource and the material to be processed.

7.2.2 *Initiate and Monitor Process Jobs*

7.2.2.1 ControlJobs are queued. ProcessJobs are not queued by equipment that supports control jobs, rather they are pooled waiting to be scheduled by their respective ControlJob. The ControlJob specifies the order for process jobs. The equipment follows that order as the equipment’s resources become available (and when material is available).

7.3 *Supplier Requirements*

7.3.1 *Management of Process Materials*

7.3.1.1 Suppliers need to implement an operational model for managing material and processing in a manner consistent with factory expectations. For instance, the equipment must know when it is finished with a carrier so that it can either allow or signal the factory for the removal of the carrier. This standard provides mechanisms to meet this requirement. While the model implies some implementation it is only the external events that are required by this standard.

7.3.2 *Control Job Events*

7.3.2.1 Control jobs supply information to host systems as either responses to request messages or as events which are sent to the host. Typically the equipment can implement the event mechanisms either in GEM (SEMI E30) or the Event Reporting standard (SEMI E53).

7.3.2.2 All state transitions defined for state models in this document must be able to be reported by separate collection events as defined in ¶6.2, State Model Methodology. The state model is the Control Job State Model (Figure 2). The data required for each state model transition event is defined per the following. This data is the minimum required per event. The host may assign other variable, as applicable, from §13, Variable Data, of this document, or from other equipment variable data.

7.3.2.3 The following data is required to be available for the Control Job State Model transition collection events:

CtrlJobID

7.4 *Operational Descriptions*

7.4.1 The ProcessJob as referenced in the specification of the control job model is the SEMI E40 process job. Within a ProcessJob the material processing order is managed by the equipment. For some equipment types, the user may be able to configure the material processing order. If available, this feature shall be fully documented by the supplier (see SEMI E40).

7.4.2 To support a simpler interface for single substrate processing, it is suggested to use the PRJobMultiCreate (see SEMI E40) service.

7.4.3 The use of control jobs restricts some SEMI E40 functionality. In particular, the equipment's queue management functionality for process jobs is super-seded by the job order as defined in the control job.

7.4.4 The relationship between control jobs and process jobs varies by equipment type. The equipment supplier should document this relationship. In general aborting or stopping a process job does not stop or abort the control job. Equipment is responsible to disposition material correctly depending on how a process job ends. In the case of equipment types that always have a one to one relationship between a control job and a process job it may be convenient for a process job abort or stop to automatically abort the respective control job. In the same sense, if a control job specifies more than one process job, it may be convenient for an abort or stop of all process jobs to automatically abort or stop the respective control job.

8 ControlJob Object Model

8.1 This specification only standardizes the ControlJob object's interface. The other objects provide a context for the ControlJob interface. Since only the interface is standardized, it is not a requirement for equipment to implement a control job object, it is only required that the equipment provide an external interface that provides the services and behavior defined for the ControlJob.

8.2 *Material to Job Linkage*

8.2.1 The equipment has relationships with many other components not illustrated in Figure 1. In particular, from its knowledge based on substrate and carrier tracking capabilities, the equipment shall connect the process job material list to the materials in the carriers that it has. It is the responsibility of the factory host to make sure that the description or identifiers of material contained in carriers can be mapped to the material identifiers in the process job definition.

8.3 *Control Jobs and Carriers*

8.3.1 A control job may specify work for several carriers. The supplier shall document the behavior of the equipment in the case where a carrier is specified for use in more than one control job.

8.4 *Attribute Definitions*

8.4.1 The attributes in Table 1 shall be accessible using the Object Services standard (SEMI E39). Object services is a set of messages which may be required of any service provider which is modeled by objects. An object model for a service provides a consistent naming convention for exchanging information between the service provider and user. Object services implementations shall be consistent with the service's object and state models. For instance, if an attribute can only be modified in a certain state, then a request to set that attribute when the model is in the wrong state shall be rejected (fail). ControlJob Attributes shall be modifiable if and only if the ControlJob is not in either the EXECUTING or COMPLETED states by using OSS to change them (see ¶14.3.1).

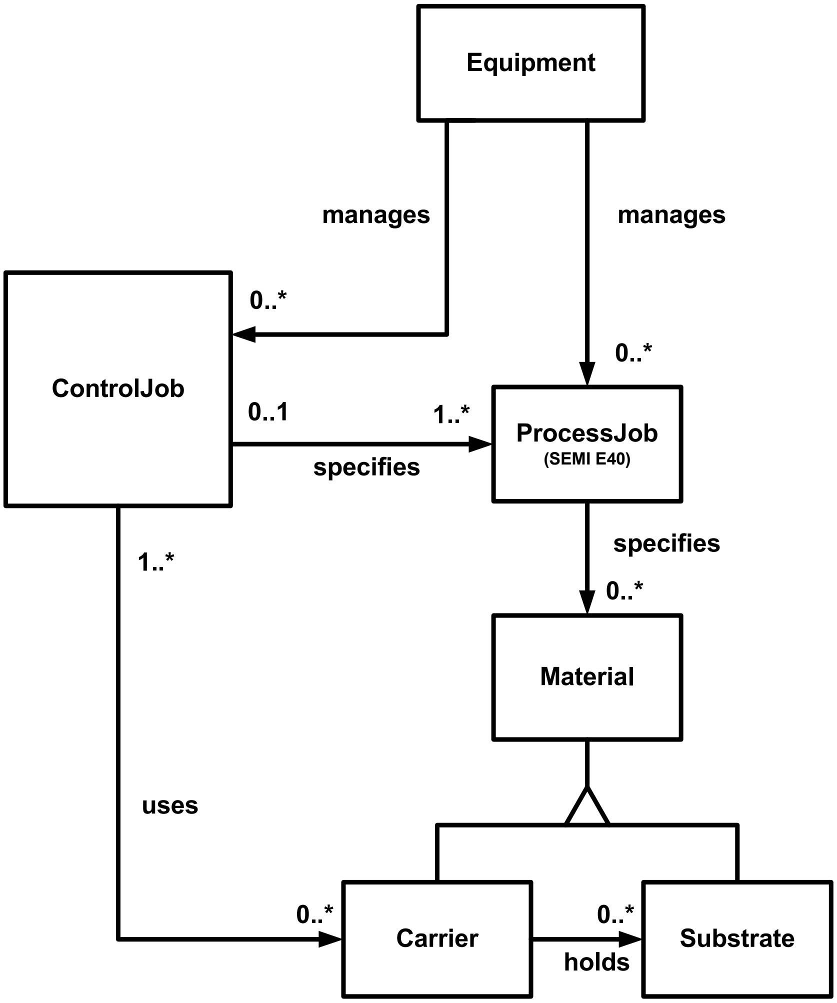

Figure 1
ControlJob Object Model

Table 1 ControlJob Attributes

Name	*Definition*	*Rqmt*	*Access*	*Form*
ObjID	Host defined identifier of the control job.	Y	RO	Text
ObjType	Object type	Y	RO	Text = 'ControlJob'
CurrentPRJob	Holds the identifiers of any process job in the active state (i.e., in the Executing, Stopping, Aborting or Pause states).	Y	RO	(list of) PRJobID (see SEMI E40)
DataCollectionPlan	Identifier for a data collection plan to be used during execution of the control job.	N	RW	Text
CarrierInputSpec	A list of carrierID for material that will be used by the ControlJob. An empty list is allowed.	Y	RW	(list of) CarrierID
MtrlOutSpec	Maps material from source to destination after processing. For uni-carrier operation, the list shall be empty. The list shall also be empty, if CarrierInputSpec is an empty list.	Y	RW	List of Structure: SourceMap DestinationMap
MtrlOutByStatus	List structure which maps locations or Carriers where processed material will be placed based on material status.	N	RW	List of structure: MaterialStatus DestinationMap

Name	*Definition*	*Rqmt*	*Access*	*Form*
PauseEvent	Identifier of a list of events on which the Control Job shall PAUSE.	N	RW	(list of) EventID
PRJobStatusList	A list of all process jobs managed by this Control Job and their associated status.	Y	RO	(list of) PRJobID PRJobState (see SEMI E40)
ProcessingCtrlSpec	A list of structures that defines the process jobs and rules for running each that will be run within this ControlJob.	Y	RW	(list of) Structure: PRJobID ControlRule OutputRule
ProcessOrderMgmt	Define the method for the order in which process jobs are initiated.	Y	RW	Enumeration: LIST ARRIVAL OPTIMIZE
StartMethod	A logical flag that determines if the ControlJob can start automatically. A user start may come through either the host connection or the operator console.	Y	RO	Boolean: TRUE – Auto FALSE – UserStart
State	The current state of the ControlJob.	Y	RO	Enumerated: per State Model

8.4.2 A number of the ControlJob attributes are composite data types. The constituent data is defined in Table 2.

Table 2 Attribute Data Definitions

Data Identifier	*Description*	*Form*
CarrierID	The identifier of a carrier that is the source or destination for substrates.	Text
ControlRule	Provides additional job control functionality. It is equipment type dependent. It may be used to modify processing based on processing results. Use of this attribute is not required for equipment, which does not support it. Suppliers shall document the use of this attribute when supported.	(list of) Structure: RuleName RuleValue
Destination	The identifier of a substrate location at which material can be placed. (Identifier should conform to standards for substrate tracking.)	Text
DestinationMap	Describes carrier positions into which finished material will be placed. If the list of carrier positions is empty, then follow sequential order of source.	Structure: CarrierID List of SubstrateLocation
MaterialStatus	ControlJob processing assigns this value to finished material. The association of MaterialStatus to Destination enables ControlJob processing to put material at the desired destination.	Equipment dependent enumeration
OutputRule	Defines the MaterialStatus (such as Good, Reject, Aborted, Monitor, etc.) based on results of the process job.	Equipment dependent
PRJobID	A process job identifier as defined by SEMI E40. Host must supply same name in the ProcessingCtrlSpec as when it requested creation of process job. NOTE: SEMI E40 process jobs link material to a recipe.	See SEMI E40.
Rule Value	The value used by the equipment for execution of a control rule.	Equipment dependent
RuleName	Identifier of a control rule.	Text
SourceMap	Describes the locations from which material is taken for processing. If the list of location is empty, then assume the default of ascending order.	Structure: CarrierID List of SubstrateLocation

Data Identifier	*Description*	*Form*
SubstrateLocation	A substrate position at a source and a destination. A carrier is an example of a multi-location destination. For a wafer carrier the SubstrateLocation is a slot number.	Numeric

8.4.3 *ControlRule*

8.4.3.1 For equipment that supports this attribute (field), the host sets this in order to achieve better host processing control capabilities. For example, the host may have previously measured characteristics of the material to be processed. A standard recipe is used based on the product and process step, but based on the measured characteristics, the application of the recipe is biased by the specified rule and the value that is passed to the rule (RuleValue). However, use of ControlRule should not be considered to be limited to only this type of application.

8.4.4 *DataCollectionPlan*

8.4.4.1 The DataCollectionPlan is a name given by the host to associate data collection activities to a specific control job. In general, it provides a way for the equip-ment to then inform and coordinate with the host to receive data collection requests. A DataCollectionPlan is generic and will be applied to many control jobs. The variable itself, DataCollectionPlan, will hold no significance for the equipment. It is simply a label the equipment reports back when requested by the host. Normally, the host upon receiving the ControlJob START event would include DataCollectionPlan as a data variable to be reported. The host then knows that the time is appropriate to set up various trace reports and event reports on the equipment. Potentially, all jobs that specify the same product type and process capability could specify the same DataCollectionPlan.

8.4.5 *OutputRule*

8.4.5.1 This attribute can only be supported by equipment that has some means to determine the status of material that it has processed. For equipment with that ability, the rule will usually take the form of a list of name value pairs. The names will be material status and the values will be measurement thresholds that correspond to the status category (such as, Good, Reject, Rework, etc.). Substrate (material) status changes should be recorded in substrate histories created by the equipment.

8.4.5.2 Equipment which also supports the MtrlOutbyStatus shall use the status determined by the OutputRule to place substrates at the Destination associated with MaterialStatus.

8.4.6 *PauseEvent*

8.4.6.1 For equipment which can support it, this attribute contains a list of equipment events, specified by the host, at which the host expects the equipment to PAUSE the ControlJob. Equipment suppliers shall document any events that can be used for the pausing of control jobs. Pausing a control job causes it to stop initiating process jobs. The host might use this to stop processing after one or more process jobs has started in order to await results before processing the rest of the material in the control job.

8.4.7 *ProcessOrderMgmt*

8.4.7.1 This is an enumerated attribute that defines the order in which processing will occur. This standard defines three enumerations. For some equipment other enumerations may be possible. If they are the supplier shall document them.

8.4.7.2 *LIST*

8.4.7.2.1 When ProcessOrderMgmt is set to this value, process jobs shall be initiated in the order of the list in ProcessingCtrlSpec.

8.4.7.3 *ARRIVAL*

8.4.7.3.1 When ProcessOrderMgmt is set to this value, process jobs shall be initiated as the material for the job arrives. Any jobs that do not require material will be initiated first.

8.4.7.4 *OPTIMIZE*

8.4.7.4.1 When ProcessOrderMgmt is set to this value, process jobs shall be initiated in an order to be determined by internal equipment algorithms, that optimize the throughput of material in the equipment.

9 Control Job State Model – Behavior

9.1 The following state chart defines the behavior of the ControlJob.

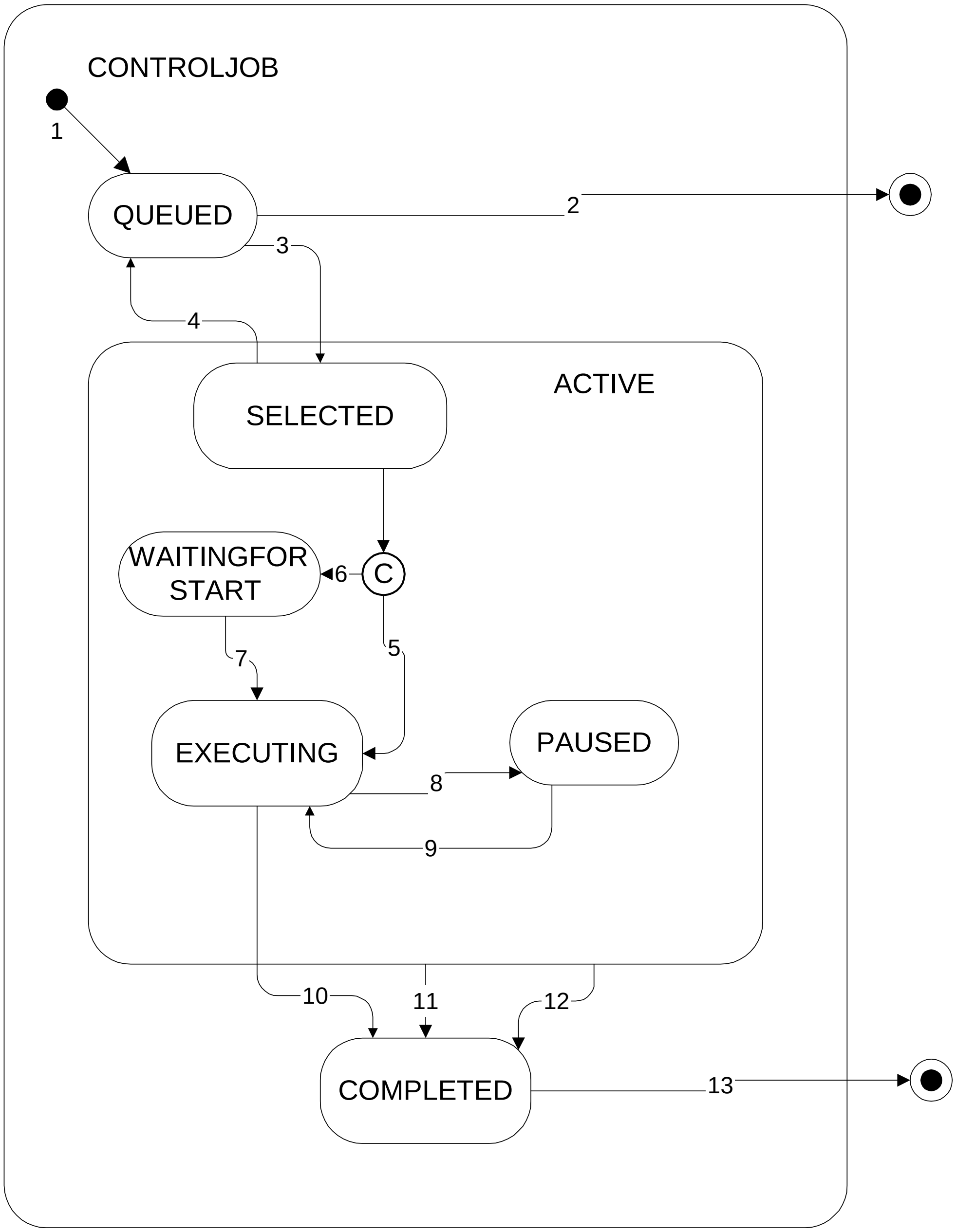

Figure 2
Control Job State Model

9.2 *State Definitions*

9.2.1 *QUEUED* — A ControlJob is queued after its creation or de-selection. A newly created ControlJob is placed at the tail of the queue.

9.2.2 *SELECTED* — In this state, the ControlJob does not initiate process jobs specified in it and therefore pre-defined (based on recipe variable parameters) process conditions can be modified. The processing resource is reserved (not available for any other jobs) by the ControlJob in the SELECTED state. If materials, required by the

ControlJob, for processing have not arrived at the equipment, the ControlJob will stay in this state until materials arrive. If the ControlJob or the first process job in the ControlJob does not require material, this state is exited immediately. A SELECTED ControlJob can be de-selected if specified materials have not arrived.

9.2.3 *WAITING FOR START* — The ControlJob is waiting to receive a start command manually or remotely from the host. The ControlJob transitions to this state only if the StartMethod is set to FALSE (UserStart) and materials have arrived.

9.2.4 *EXECUTING* — In this state, each process job in the ProcessingCtrlSpec is initiated in order, based on the value of the ControlJob's ProcessOrderMgmt attribute as required resources become available and material for the job has been verified. Process jobs that have been initiated but that are WAITINGFORSTART or PAUSED shall block the availability of the resources that they require (see SEMI E40).

NOTE 1: Process jobs that have blocked available resources shall cause the ControlJob to stop initiating subsequent process jobs that use those resources.

9.2.5 *PAUSED* — When the ControlJob is paused it shall not commence the initiation of any more Process jobs. In this state, Process jobs that have not entered the "PROCESSING" state can be modified. Various attributes of the ControlJob can also be modified. This is equipment specific and shall be documented by the equipment supplier.

9.2.6 *COMPLETED* — A control job enters this state once all of its process jobs have been completed, stopped or aborted.

Table 3 ControlJob State Transition Table

Num	*Previous State*	*Trigger*	*New State*	*Actions*	*Comments*
1	(No state)	Receive "Create" command from host or operator through operator console.	QUEUED	Create ControlJob and put it at the tail of a control job queue.	If job queue is full, "Create" request is rejected.
2	QUEUED	Receive "Cancel", "Abort", or "Stop" command from host or operator through operator console.	(No state)	De-queue and terminate the job. Send a "ControlJob Canceled" event to the host.	If other control jobs are waiting behind the canceled job in the queue, they are shifted forward to fill in the gap after the de-queuing of the canceled control job.
3	QUEUED	The processing resource has capacity to begin work on the next ControlJob.	SELECTED	Select and de-queue the job at the head of the queue. Send a "Selected" event to the host.	Materials are not necessarily at the equipment.
4	SELECTED	Receive "De-select" command from host or operator through operator console and materials for the control job have not arrived yet.	QUEUED	De-selected job moves to the head of the job queue and the job that was at the head becomes the SELECTED job.	The command shall be rejected if the resources for the job at the head of queue are not available. See the Queue Model.
5	SELECTED	Material for the first process job arrives or in the case where the first (or only) process job does not require material, this transition shall be taken as soon as the processing resource for that process job becomes available. "StartMethod" attribute in the ControlJob is set for Auto.	EXECUTING	Send "Execution began" event to the host.	Process jobs associated with a carrier will not initiate until the identifier and substrate slot map for the carrier have been verified. Process jobs that don't use material can be initiated immediately.
6	SELECTED	Same as for transition 5 except that the "StartMethod" attribute in the control job is set for user start.	WAITING FORSTART	Send a "JobWaiting for Start" event to host and/or operator.	

Num	*Previous State*	*Trigger*	*New State*	*Actions*	*Comments*
7	WAITING FORSTART	User START command received.	EXECUTING	Same as for transition 5.	Same as for transition 5.
8	EXECUTING	Received "Pause" message from host or operator through operator console or a ControlJob. PauseEvent has occurred.	PAUSED	Send a "Paused" event to the host.	Process jobs which have not started can be modified in this state.
9	PAUSED	Receive "Resume" message from host or operator through operator console.	EXECUTING	Commence initiating process jobs. Send a "Resumed" event to the host.	
10	EXECUTING	All the ProcessJobs specified for the ControlJob have completed.	COMPLETED	Send a "Complete" event to the host.	It may include post processing completion.
11	ACTIVE	Receive "CJStop" message from host or operator through operator console and all the process jobs under the ControlJob have been stopped and material processing is stopped.	COMPLETED	Send a "Stopped" event to the host.	
12	ACTIVE	Receive "CJAbort" command from host or operator through operator console and all the process jobs under the ControlJob have been aborted and material processing is aborted.	COMPLETED	Send "Aborted" message to the host.	
13	COMPLETED	The ControlJob is deleted.	(No state)		Equipment should perform this function automatically for COMPLETED jobs.

10 Control Job Queue Model

10.1 The Queuing mechanism for control jobs will generally operate under FIFO (First in- First Out) constraints. The commands used to monitor queue status and prevent deadlock conditions are specified here.

10.2 *Queue Integrity*

10.2.1 To maintain queue integrity, only one operation shall be performed at any given time (e.g., the "Create" request shall be rejected by the equipment if the CJHOQ service is being processed). The Queue is defined to be "locked" while it is performing an operation (and refusing any further operation until completion of the current operation).

10.3 *Head of Queue Service*

10.3.1 The Head of Queue service (CJHOQ) shall operate under the following rules:

1) All control jobs positioned between the specified control job and the head of the queue (including the job currently positioned at the head) will be moved back one position. The specified control job will then be moved into the head of queue position.

2) When the CJHOQ command is invoked, the queue will be "locked" to maintain integrity.

3) In the case where only one control job exists in the queue, the command will perform no action on the queue.

10.4 *DeadLocks*

10.4.1 The Head of Queue service (CJHOQ) requests a specific control job to be set as the next control job to be run. In order to prevent deadlock when the job at Head of Queue and the job in the SELECTED state are both awaiting material delivery, the CJHOQ command may be used to move a different job to the head of the queue position. The potential dead lock is then broken by issuing the DE-SELECT service request.

10.5 *Utilization of Queue for Control Job Priority Management*

10.5.1 Similar to Deadlock, certain cases may arise where a job at the head of the queue cannot be selected due to a lack of processing resources. In this case, DE-SELECT command shall be rejected and consequently a series of DE-SELECT and CJHOQ commands may be issued in an attempt to find a job which can transition to SELECTED state.

10.5.2 To force a "hot job" to be the next job run, it could be necessary to send the CJStop message to the SELECTED job. This case only happens when the job at the head of the queue (the hot job) does not yet have resources available.

NOTE 2: Most equipment where this is possible can support parallel execution of Control Jobs. Management of the queue(s) in this case is currently beyond the scope of this standard.

10.6 *Space in the Queue*

10.6.1 The QueueAvailableSpace variable data item is used to query the number of control job openings within the queue. The QueueAvailableSpace shall function according to the following rules:

1) This variable can only be guaranteed valid when no other operations are being performed simultan-eously on the queue. For example, don't request this variable while a Create control job command is being processed by the equipment.

2) This variable shall be incremented whenever a control job in the queue is de-queued. That is, when a "Cancel", "Abort", or "Stop" command has been received and completed while the control job is queued. It should also be incremented when the SELECTED state is entered by a control job. However, if this transition occurs as a result of the "Deselect" command (at least one job in queue, and a control job in the SELECTED state), no change should be made to the variable value.

3) This variable should be decremented whenever a control job joins the queue. That is, when a "Cre-ate" command is received and accepted. It should also be decremented if a "Deselect" command is issued on a control job in the SELECTED state and no other jobs currently reside in the queue.

4) The equipment should reject the "Create" command when this variable is equal to zero.

10.7 *Getting a List of Queued Jobs*

10.7.1 The QueuedCJobs Variable Data Item is used to query the names of the control jobs currently residing in the queue. It lists items starting at the head of the queue.

10.7.2 The QueuedCJobs Variable Data Item shall function according to the following rules:

1) This variable can only be guaranteed valid when no other operations are being performed simultan-eously on the queue.

2) This variable list will be modified whenever a suc-cessful "Create" command is received. Addition-ally, "Cancel", "Abort", and "Stop" commands issued on control jobs residing in the queue will modify this variable list. Any use of the "Deselect" command will also modify the variable list.

11 Properties for Carriers

11.1 Compliance to Control Job Management requires that the equipment track the status of individual carriers. In particular, the factory needs to know the status of carriers with respect to control jobs. Carriers may have various properties that are beyond the scope of this standard. However, there are specific properties that are needed for Control Jobs.

11.2 *Carrier Verification*

11.2.1 The equipment needs to know when a carrier has been verified as proper. Only process jobs associated with the carrier that has been verified shall be initiated by a control job. Depending on the equipment's capabilities, verification may include verification by an equipment read of the carrier's ID (identification) and the reading of the substrate (e.g., wafer) slot map. The ControlJob determines if the carrier is verified by checking the carrier's attributes that indicate the level to which a carrier has been verified. This section is reserved for further specification of this requirement.

11.3 *Carrier Completion for Control Jobs*

11.3.1 A carrier that has been loaded onto the equipment may go through three stages: first, it is in the "not accessed" stage until it is accessed by a control job. When the carrier is at the substrate port and unloading of the substrates within the carrier begins, the carrier enters the "in access" stage. Once it enters this stage, it remains in this stage until all material has been returned to the carrier, no active control job exists that is using it and no control jobs in the queue exist that reference it. The equipment shall provide a property of the carrier that shows the current stage of all carriers. The equipment shall provide a "CarrierComplete" or "CarrierStopped" event for each change in stage. The event when the carrier enters the "in access" stage informs the host that the carrier may not be removed from the equipment. The event when the carrier enters the "completed" stage informs the host that the carrier may be removed.

12 Requirements - Service Definitions

12.1 *Service Definitions*

Table 4 Service Definitions Table

Message Service Name	*Type*	*Description*
CJStart	R	To start a ControlJob.
CJPause	R	To request a ControlJob to pause.
CJResume	R	To request a PAUSED ControlJob to go to the EXECUTING state.
CJCancel	R	To request a ControlJob to be removed from the queue.
CJDeselect	R	To request a ControlJob to be deselected; it will no longer be the next job to run.
CJStop	R	To request a ControlJob to stop. Used to discontinue a job without risk to the material.
CJAbort	R	To request a ControlJob to abort. Used to discontinue a job on equipment that may be malfunctioning. Material is at risk when this command is issued.
CJHOQ	R	To request a particular ControlJob to be set as the next control job to be SELECT'ed.

12.2 *Parameter Definitions*

Table 5 Parameter Definitions Table

Parameter Name	*Description*	*Format: Possible Value*
ACKcode	To return indication of result of service call.	Enumeration: SUCCESS, FAILURE
Action	See ¶12.2.1	Enumeration: SAVEJOBS REMOVEJOBS
CtrlJobID	ObjID (object identifier) of a control job.	Text

Parameter Name	*Description*	*Format: Possible Value*
ErrorCode	Contains the code for the specific error found.	Enumerated (ACKcode must equal FAILURE): *All services:* • Unknown object instance • Parameters improperly specified • Insufficient parameters specified *CJStart:* • Command not valid for current state *CJPause:* • Command not valid for current state *CJResume:* • Command not valid for current state *CJCancel:* • Job cancelled • Command not valid for current state *CJDeselect:* • Command not valid for current state • Busy (when queue empty or resources for HOQ job would not be available) *CJStop:* • Job stopped *CJAbort:* • Job aborted *CJHOQ:* • Command not valid for current state
ErrorInfo	The parameter may be null or excluded on a SUCCESS.	(List of) ErrorCode ErrorText
ErrorText	Description of the error.	Text
Status	Information returned by service provider which indicates the result of the service call.	Structure: ACKcode ErrorInfo

12.2.1 *Action Parameter* — The Control Job services CJCancel, CJStop, and CJAbort can all cause Process Jobs to be terminated, as specified in SEMI E40. In these Control Job services, the parameter “Action” specifies the disposition of the Process Job objects for the terminated Process Jobs. “Action” specifies one of the following enumerated values: SAVEJOBS and REMOVEJOBS. This Action parameter only affects Process Jobs that are in the QUEUED/POOLED state.

12.3 *Message Details*

12.3.1 This section specifies parameter usage by the service messages.

12.3.2 *Creating ControlJobs* — ControlJobs shall be created by using the OSS (SEMI E39) Object Create message. The following table defines the use of the AttrSetting arguments to the Object Create service. Note: ObjType is a required argument of Object Create and therefore should not be reset by including it as an AttrSetting argument. In the table M indicates mandatory, O indicates optional, and R specifies restricted (shall be ignored if used).

12.3.3 The process jobs specified for a control job must exist prior to calling this message. If a process job does not exist, the Create service shall fail and the ObjStatus shall return a list of any PRJob identifiers that were not present. The Create request shall be rejected if the ControlJob queue is full. Newly created control jobs will be put at the end of the queue.

Table 6 SetAttr Arguments Table

Control Job Attribute Name	*Use as AttrSetting of Create Service*
ObjID	M
ObjType	R
CurrentPRJob	R
DataCollectionPlan	O
CarrierInputSpec	M
MtrlOutSpec	M
MtrlOutbyStatus	O
PauseEvent	O
ProcessingCtrlSpec	M
ProcessOrderMgmt	M
StartMethod	M
State	R

12.3.4 *CJStart* — Starts jobs that require a user start. The host sends this command only to a Control Job from which it has received a WAITINGFORSTART event.

Table 7 CJStart Service Parameter Definitions Table

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
CtrlJobID	M	M	Indicate which job
Status	-	M	Success or failure

12.3.5 *CJPause* — The ControlJob shall stop initiating process jobs. Process jobs in the EXECUTING state are not affected by this command.

Table 8 CJ Pause Service Parameter Definitions Table

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
CtrlJobID	M	M	Indicate which job
Status	-	M	Success or failure

12.3.6 *CJResume* — The ControlJob shall resume initiating process jobs.

Table 9 CJResume Service Parameter Definitions Table

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
CtrlJobID	M	M	Indicate which job
Status	-	M	Success or failure

12.3.7 *CJCancel* — Used to remove a ControlJob from the Queue. The command shall only succeed for jobs in the QUEUED state.

Table 10 CJCancel Service Parameter Definitions Table

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
CtrlJobID	M	M	Indicate which job
Action	M	-	
Status	-	M	Success or failure

12.3.8 *CJDeselect* — Shall only succeed for jobs in the SELECTED state. Deselected jobs must trade places with the job that is currently at the head of the queue. If the job at the head of the queue cannot transition to the SELECTED state, then the deselect request shall be rejected. See the section on Control Job Queue Model for information on breaking possible deadlocks.

Table 11 CJDeselect Service Parameter Definitions Table

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
CtrlJobID	M	M	Indicate which job
Status	-	M	Success or failure

12.3.9 *CJStop* — Stops the ControlJob from initiating any more process jobs. Equipment should issue a STOP command to all running process jobs. When the currently running process jobs have stopped, the ControlJob will send a complete event with a status code indicating the ControlJob has stopped. ControlJobStop shall only succeed on a job in the ACTIVE or QUEUED states. When CJStop is issued in the QUEUED state, its affect will be identical to that of CJCancel.

Table 12 CJStop Service Parameter Definitions Table

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
CtrlJobID	M	M	Indicate which job
Action	M	-	
Status	-	M	Success or failure

12.3.10 *CJAbort* — Stops the control job from initiating any more process jobs. The currently running process jobs are sent the Abort command by the equipment. When the equipment has detected the successful ABORT of currently running process jobs, the ControlJob shall send a complete event with a status code indicating the job was aborted. ControlJobAbort shall only succeed on a job in the ACTIVE or QUEUED states. When CJAbort is issued in the QUEUED state, its affect will be identical to that of CJCancel. The equipment due to a serious alarm situation (operator risk) may internally generate this command.

Table 13 CJAbort Service Parameter Definitions Table

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
CtrlJobID	M	M	Indicate which job
Action	M	-	
Status	-	M	Success or failure

12.3.11 *CJHOQ* — The other jobs in the queue are pushed back (rest of queue order remains unchanged).

Table 14 CJHOQ Service Parameter Definitions Table

Parameter	*Req/Ind*	*Rsp/Cnf*	*Comment*
CtrlJobID	M	M	Indicate which job
Status	-	M	Success or failure

13 Variable Data

13.1 For objects defined by Control Job Management, the identifiers of the objects and all of the attributes of the objects shall be available for inclusion in event reports associated with those objects. The following attribute is the most likely to be used: CtrlJobID.

13.2 The following table provides the definition of additional Variable Data that equipment shall support for Control Job Management.

Table 15 Variable Data Definitions Table

Variable Name	*Description*	*Type*	*Access*	*Comment*
CtrlJobID	Control job identifier, available to be used in control job related event reports.	Text	RO	
QueuedCJobs	This is an ordered list of control jobs currently in the Queue. The first job in the list is the job at the head of the Queue.	(list of) Text	RO	Each list item is a control job identifier.
QueueAvailableSpace	Indicates number of jobs which the Queue can accept.	Numeric	RO	This value cannot be negative. When it is zero it indicates that the queue is full.
SetUpName	Host sets this to define the operational condition of the equipment.	Text	RW	If the equipment is manipulated locally, it should set this variable to "unknown", otherwise it returns the value set by the host (when requested).

14 Additional Requirements

14.1 *Serial Execution of Control Jobs*

14.1.1 Control jobs are initiated in sequential order by the equipment. The order is based on the queue. A ControlJob shall not issue a complete event (message) until all substrates have been placed in destination carriers. However, in many cases equipment must support multiple control jobs running at once; in order to support the factory requirement for equipment productivity. This will be particularly true for multi-module equipment. The next ControlJob in the queue shall start as soon as possible after processing has begun for the last ProcessJob in the previous ControlJob's processing control specification.

14.2 *Parallel Execution of Control Job*

14.2.1 Some equipment may be able to support parallel execution of control jobs. The supplier must fully document this behavior and any additional services needed to manage it.

14.3 *Modifying Control Jobs*

14.3.1 Control jobs shall be modifiable if and only if they are not in either the EXECUTING or COMPLETED states. Jobs shall be modified by using OSS to change their attributes. Modifications shall be rejected if the equipment is in the wrong state or requested value changes are out of range.

14.4 *Set-up, Pre- and Post-Conditioning*

14.4.1 Whenever equipment has completed some processing work, with or without material, the equip-ment can be considered to be "set-up" for a certain process capability. Information about the equipment's set-up is important to the factory in determining the best material routing. The SetupName variable defined in this standard is set by the host after host directed processing or changes to equipment constants. If the equipment is used for processing while off-line or not under host command, the value of the variable shall be set to "unknown". It shall also be set to "unknown" immediately after any changes to equipment constants.

14.5 *Event Relationships*

14.5.1 This section is reserved for specification of the relationship between process job events and control job events.

14.5.2 PRJob Paused

14.5.3 PRJob Aborted or Stopped

15 Compliance

15.1 Implementations compliant to this standard shall implement all the messages as specified in §12. All mandatory parameters must be supported. The supplier shall document support for any of the optional parameters. Any additional parameters and messages shall be fully documented by the supplier. Additional messages shall be used to support additional functionality and not as a replacement for any of messages specified herein.

15.2 Table 16 provides a checklist for Control Job Mangement (CJM) compliance.

Table 16 CJM Compliance Statement

Fundamental CJM Requirements	*CJM Section*	*Implemented*	*CJM Compliant*
Control Job Object	8	□ Yes □ No	□ Yes □ No
Control Job State Model	9	□ Yes □ No	□ Yes □ No
Control Job Queue Model	10	□ Yes □ No	□ Yes □ No
Carrier Properties	11	□ Yes □ No	□ Yes □ No
Service Message Implementation	12	□ Yes □ No	□ Yes □ No
Variable Data	13	□ Yes □ No	□ Yes □ No
Events	7.3.2	□ Yes □ No	□ Yes □ No
Additional Requirements	14	□ Yes □ No	□ Yes □ No
Additional CJM Capabilities	*CJM Section*	*Implemented*	*CJM Compliant*
none			

RELATED INFORMATION 1
APPLICATION NOTES

NOTICE: The material contained in these Applications Notes is not an official part of SEMI E94 and is not intended to modify or supersede the official standard. Rather, these notes are auxiliary information describing possible methods for implementing the protocol described by the standard and are included as reference material. The standard should be referred to in all cases. SEMI makes no warranties or representations as to the suitability of the material set forth herein for any particular application. The determination of the suitability of the material is solely the responsibility of the user.

Service messages are presented with a "C" language style of application interface. For illustrative purposes values for variables may be included using the "=" in the argument list. If not used the optional arguments are not shown. Service names (e.g., PRJob) are prefixed to function name. The scenarios assume processing is for wafers. The services used in these scenarios are PRJob for Processing Management, ControlJob for Control Job Management, and CMS for Carrier Management Services.

R1-1 ControlJob for a Batch Processing Tool

R1-1.1 Tool processes the contents of a single carrier as a batch. This example demonstrates the simplicity of using control jobs for a simple situation. Wafer order is maintained, and material is returned to the source carrier.

#	*Comment*	*Dir*	*Message*	*CJS*	*PJS*
1	Create the process job.	H->E	PRJobCreateEnh (PRJobID=prj01_04, Mtrl = CS001, RecID=ILD3)	No state	No state
2		H<-E	PRJobCreateAck (PRJobID, PRJobStatus)		In POOL
3	Request a control job. Material out specification maintains wafer order from source carrier to destination carrier.	H->E	ControlJobCreate (CtrlJobID=cjf01_01, ProcessingCtrlSpec= (prj01_04, null,null), MtrlOutSpec=(CSA01,null,null), MaterialIn= CS001, StartMethod=AUTO)		
4	Request accepted.	H<-E	ControlJobCreateAck (CtrlJobID=cjf01_01, JobStatus)	QUEUED	
5	No CJ in selected state so the newly created job immediately transitions.	H<-E	Event (CJSELECTED, CtrlJobID=cjf01_01)	SELECTED	
6	A carrier for the selected control job arrives.	H<-E	Event (CARRIERIDREAD, CID=CS001)		
7	The equipment recognizes it and starts execution of the ControlJob.	H<-E	Event (ControlJobStart, cjf01_01)	EXECUTING	
8	ControlJob starts the ProcessJob; begins loading wafers to the processing boat.	H<-E	Event (PRJOBSETUP, prj01_01)		ACTIVE/ SETUP
9	Material processing starts after all material is in the processing boat.	H<-E	Event (PRJOBPROCESSING, prj01_04)		ACTIVE/ PROCESSING
10	Equipment begins to return material to the source carrier (= destination carrier).	H<-E	Event (PROCESSINGCOMPLETE, PRJob=prj01_04)		ACTIVE/ PROCESSING COMPLETE
11	Carrier is filled with wafers.	H<-E	Event (CarrierComplete=CS001)		
12	Host wants to get it.	H->E	Rcommand (CarrierOut=CS001)		
13	Equipment indicates carrier can now be picked up.	H<-E	Event (ReadytoUnload, CarrierID=CS001, PortID)		
14		H<-E	Event (PRJOBCOMPLETE, PRJob=prj01_01,)		No state

#	*Comment*	*Dir*	*Message*	*CJS*	*PJS*
15	Jobs may complete before the carrier is picked up.	H<-E	Event (ControlJobCompleted=cjf01_01, status=OK)	COMPLETE	

R1-2 ControlJob for a Single Wafer Processing Tool

R1-2.1 Will be added later.

R1-3 ControlJob for Single Wafer Processing with Recipe Variable Parameters

R1-3.1 Will be added later.

R1-4 Error Recovery of Batch Tool ControlJob, Carrier Slot Map Failure

R1-4.1 Will be added later.

R1-5 Carrier Swap During Processing

R1-5.1 Multiple carriers are loaded to a batch processing tool that has buffering, after carriers are emptied, they are removed and new empty carriers are loaded. In this example, it requires four carriers for a batch.

CJS = Control Job State, PJS = Process Job State

#	*Comment*	*Dir*	*Message*	*CJS*	*PJS*
1	Create the process job.	H->E	PRJobCreateEnh (PRJobID=prj01_01, Mtrl = CSA01,CSA02, CSA03, CSA04, RecID=ILD1)	No state	No state
2		H<-E	PRJobCreateAck (PRJobID, PRJobStatus)		In POOL
3	Request a control job. Material out specification maintains wafer order from source carrier to destination carrier.	H->E	ControlJobCreate (CtrlJobID=cjf01_01, ProcessingCtrlSpec= (prj01_01, null,null), ProcessOrderMgmt = ARRIVAL, MtrlOutSpec=(((CSA01,null),(CSB01, null)), ((CSA02, null), (CSB02,null)), ((CSA03, null), (CSB03,null)), ((CSA04, null), (CSB04,null))), MaterialIn= (CSA01,CSA02, CSA03, CSA04), StartMethod=AUTO)		
4	Request accepted.	H<-E	ControlJobCreateAck (CtrlJobID=cjf01_01, JobStatus)	QUEUED	
5	No CJ in selected state so the newly created job immediately transitions.	H<-E	Event (CJSELECTED, CtrlJobID=cjf01_01)	SELECTED	
6	A carrier for the selected control job arrives.	H<-E	Event (CARRIERIDREAD, CID=CSA01)		
7	The equipment recognizes it and starts execution of the ControlJob.	H<-E	Event (ControlJobStart, cjf01_01)	EXECUTING	
8	ControlJob starts the ProcessJob; begins loading wafers to the processing boat.	H<-E	Event (PRJOBSETUP, prj01_01)		ACTIVE/SETUP
9		H<-E	Event (CarrierEmpty,CSA01)		
10	The next carrier arrives.	H<-E	Event (CARRIERIDREAD, CSA02)		
11		H->E	Rcommand (CarrierOut=CSA01)		
	As carriers are emptied they are removed.		Steps 9 through 11 are repeated 3 more times		

#	Comment	Dir	Message	CJS	PJS
12	Material processing starts after all material is in the processing boat.	H<-E	Event (PRJOBPROCESSING, prj01_01)		ACTIVE/ PROCESSING
13	Output carriers begin to arrive.	H<-E	Event (CARRIERIDREAD, CID=CSB01)		
	Until all output carriers arrive.		Step 13 is repeated 3 more times		
14	Equipment begins to load output carriers.	H<-E	Event (PROCESSINGCOMPLETE, PRJob=prj01_01)		ACTIVE/ PROCESSING COMPLETE
15	First output carrier is filled with wafers.	H<-E	Event (CarrierComplete=CSB04)		
16	Host wants to get it.	H->E	Rcommand (CarrierOut=CSB04)		
17	Gets rest of carriers.		Steps 15 and 16 are repeated 3 more times		
18		H<-E	Event (PRJOBCOMPLETE, PRJob=prj01_01,)		No state
19		H<-E	Event (ControlJobCompleted=cjf01_01, status=OK)	COMPLETE	

R1-6 Using Cleaning Wafers

R1-6.1 For this scenario we assume a single wafer processing tool, such as an RIE. The tool has three fixed load ports; two for product material, one for cleaning material. A cleaning wafer is run before the 1st and 13th wafer of each carrier of product wafers. Show the load and unload of the cleaning wafers and the running of the ControlJob for processing material. Note that control jobs do not provide functionality for dispositioning collateral material consumed during processing (the cleaning wafers). The equipment is responsible for providing mechanisms to determine when this collateral material has been consumed (should be replaced).

Step #	Comment	Dir	Message	CJS	PJS
1	Remove the previously spent wafers.	H<-E	Event (ReadytoUnload, PortID=Cleaning, CarrierID=CC01)	No state	No state
2	After host picks up the spent wafers, the equipment is ready to load.	H<-E	Event (ReadytoLoad, PortID=Cleaning)		
3	Delivered material is identified.	H<-E	Event (CarrierIDRead, CarrierID=CC02)		
4	Host directs carrier to be moved to the wafer access position.	H->E	CMSProceedwithCarrier (CarrierID=CC02)		
5	Create a job to run a cleaning wafer.	H->E	PRJobCreate (PRJobID=prj01_01, Mtrl = ACleanWafer, RecID=CleaningProcess)		
6		H<-E	PRJobCreateAck (PRJobID, PRJobStatus)		In POOL
7	Create the jobs for the first 12 wafers in the carrier (PW1-12).	H->E	PRJobDuplicateCreate (PRJobSpecList = (prj01_02, PW1), (prj01_03, PW2), …(prj01_13, PW12), RecID = P64ME5, Start=AUTO, MtrlType=WAFER)		
8		H<-E	PRJobCreateAck (PRJobIDList, PRJobStatus)		In POOL
9	Another cleaning wafer	H->E	PRJobCreate (PRJobID=prj01_14, Mtrl = ACleanWafer, RecID=CleaningProcess)		
10		H<-E	PRJobCreateAck (PRJobID, PRJobStatus)		In POOL

Step #	*Comment*	*Dir*	*Message*	*CJS*	*PJS*
11	The rest of the product wafers	H->E	PRJobDuplicateCreate (PRJobSpecList = (prj01_15, PW13), (prj01_16, PW14), …(prj01_27, PW25), RecID = P64ME5, Start=AUTO, MtrlType=WAFER)		
12		H<-E	PRJobCreateAck (PRJobIDList, PRJobStatus)		In POOL
13	Now the control job	H->E	CtrlJobCreate (CtrlJobID=cj01_01, ProcessingCtrlSpecList = (prj01_01, null, null), (prj01_02, null, null), …(prj01_27, null, null), CarrierInputSpec = CP01, Start = AUTO, ProcessOrderMgmt = LIST, MtrlOutSpec = (CP01, null, CP01,null)		
14		H<-E	CtrlJobCreateAck (ID = cj01_01, Status=OK)	QUEUED -> SELECTED	
15		H<-E	Event (ReadytoLoad, PortID = P1)		
16		H<-E	Event (CarrierIDRead, CarrierID=CP01)		
17		H->E	CMSProceedwithCarrier (CP01)		
18		H<-E	Event (CtrlJobStart = cj01_01)	EXECUTING	
	Host might not even have these sent.		Lots of PRJob Start and End Events		
19		H<-E	Event (CtrlJobComplete = cj01_01)	COMPLETED	
20		H<-E	Event (ReadytoUnload, CarrierID = CP01)		

NOTICE: SEMI makes no warranties or representations as to the suitability of the standard set forth herein for any particular application. The determination of the suitability of the standard is solely the responsibility of the user. Users are cautioned to refer to manufacturer's instructions, product labels, product data sheets, and other relevant literature respecting any materials mentioned herein. These standards are subject to change without notice.

The user's attention is called to the possibility that compliance with this standard may require use of copyrighted material or of an invention covered by patent rights. By publication of this standard, SEMI takes no position respecting the validity of any patent rights or copyrights asserted in connection with any item mentioned in this standard. Users of this standard are expressly advised that determination of any such patent rights or copyrights, and the risk of infringement of such rights, are entirely their own responsibility.

Copyright by SEMI® (Semiconductor Equipment and Materials International), 3081 Zanker Road, San Jose, CA 95134. Reproduction of the contents in whole or in part is forbidden without express written consent of SEMI.

SEMI E94.1-1104
SPECIFICATION FOR SECS-II PROTOCOL FOR CONTROL JOB MANAGEMENT (CJM)

This specification was technically approved by the Global Silicon Wafer Committee and is the direct responsibility of the North American Silicon Wafer Committee. Current edition approved by the North American Regional Standards Committee on August 16, 2004. Initially available at www.semi.org September 2004, to be published November 2004. Previously published November 2001.

1 Purpose

1.1 This document maps the services and data of SEMI E94 to SECS-II streams and functions and data definitions.

2 Scope

2.1 This is a specification covering equipment supporting automated control job management.

2.2 This document applies to all implementations of SEMI E94 that use the SECS-II message protocol (SEMI E5). Compliance to this standard requires compliance to both SEMI E94 and SEMI E5.

NOTICE: This standard does not purport to address safety issues, if any, associated with its use. It is the responsibility of the users of this standard to establish appropriate safety and health practices and determine the applicability of regulatory or other limitations prior to use.

3 Limitations

3.1 This specification applies to semiconductor equipment that also uses SEMI E40 Process Jobs.

4 Referenced Standards

4.1 *SEMI Standards*

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

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

SEMI E39.1 — SECS-II Protocol for Object Services Standard (OSS)

SEMI E40 — Standard for Processing Management (PJM)

SEMI E53 — Event Reporting (ER)

SEMI E94 — Specification for Control Job Management (CJM)

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

5 Terminology

5.1 None.

6 Mapping

6.1 This section shows the specific SECS-II streams and functions that shall be used for SECS-II implementation of the services defined in SEMI E94, as well as the parameter mapping for data attached to services.

6.2 *Message Mapping*

6.2.1 *Services Message Mapping*

6.2.1.1 Table 1 defines the relationships between SEMI E94 services and SECS-II messages.

Table 1 Services Message Mapping Table

Service Name	*Stream, Function*	*SECS-II Message Name*
Create Object	S14,F9/10	Create Object Request/Acknowledge (SEMI E39.1)
CJAbort	S16,F27/28	Control Job Command Request/Acknowledge
CJCancel	S16,F27/28	Control Job Command Request/Acknowledge
CJDeselect	S16,F27/28	Control Job Command Request/Acknowledge
CJHOQ	S16,F27/28	Control Job Command Request/Acknowledge
CJPause	S16,F27/28	Control Job Command Request/Acknowledge
CJResume	S16,F27/28	Control Job Command Request/Acknowledge
CJStart	S16,F27/28	Control Job Command Request/Acknowledge
CJStop	S16,F27/28	Control Job Command Request/Acknowledge

6.2.2 *Event Message Mapping*

6.2.2.1 Table 2 defines the relationships between SEMI E94 collection events and SECS-II messages.

Table 2 Event Message Mapping Table

Event Name	*Stream, Function*	*SECS-II Message Name*
All state model transitions	If SEMI E30 style events: S6,F11/12 If SEMI E53 style events: S6,F11/12 S6,F13/14	If SEMI E30 style events: Event Report Send/Acknowledge If SEM E53 style events: Event Report Send/Acknowledge Annotated Event Report Send/Ack

6.2.3 *Parameter Mapping*

6.2.3.1 Table 3 defines the relationships between SEMI E94 service parameters and SECS-II data definitions.

Table 3 Parameter to SECS-II Data Items Mapping

Parameter Name	*Range*	*SECS-II Data Item*
ACKcode	TRUE, FALSE	ACKA
Action	SAVEJOBS, REMOVEJOBS	CPVAL (U1) 0 = SAVEJOBS. This command does not destroy the Process Jobs specified by this Control Job. 1 = REMOVEJOBS. This command destroys all Process Jobs specified by this Control Job.
CtrlJobID	1–80 characters (Conforms to ObjID in SEMI E39.1, Section 6.)	OBJID
ErrorCode	Enumerated	ERRCODE
ErrorInfo	Error	L,2 1. ErrorCode 2. ErrorText
ErrorText	1–80 characters	ERRTEXT
Status	Acknowledgement and error	L,2 1. ACKcode 2. ErrorInfo

6.2.4 *SECS-II Data Items without Corresponding SEMI E94 Parameters*

6.2.4.1 Table 4 contains the SECS-II data items that do not correspond to SEMI E94's service parameter.

Table 4 Additional Data Item Requirements Table

Function	*SECS-II Data Item*
Used to identify control job commands.	CTLJOBCMD
Used to satisfy SECS-II conventions for linking a multi block inquiry with subsequent multi block message.	DATAID

6.2.5 *Variable Data Item Mapping*

6.2.5.1 Table 5 shows the specific SECS-II data classes, and formats needed for SECS-II implementations of SEMI E94 variable data items.

Table 5 Variable Data Item Mapping

Variable Name	*Class*	*Format*
CtrlJobID	DVVAL	<OBJID>
QueueAvailableSpace	SV	5() (U1, U2, U4, U8)
QueuedCJobs	SV	L,n n = number of queued control jobs 1. <CtrlJobID$_1$> … n. <CtrlJobID$_n$>
SetUpName	ECV	20 (A[1..80]) Zero length string is used by the equipment to indicate "unknown".

7 SECS-II Attribute Definitions

7.1 *ControlJob Object SECS-II Attributes Definitions*

7.1.1 The following are the SECS-II structure definitions for the E94 ControlJob Object.

Table 6 ControlJob Object Attribute Definitions

Attribute Name	*Attribute Data Form: SECS-II Structure*
"ObjType"	20 (A) "ControlJob"
"ObjID"	<OBJID> CtrlJobID (Conforms to the restrictions of ObjID as specified in SEMI E39.1, Section 6.)
"CarrierInputSpec"	L,n n = number of input carriers 1. <CARRIERID$_1$> … n. <CARRIERID$_n$>
"CurrentPRJob"	L,n n = number of process jobs 1. <PRJOBID$_1$> … n. <PRJOBID$_n$>
"DataCollectionPlan"	20 (A) DataCollectionPlan

Attribute Name	*Attribute Data Form: SECS-II Structure*
“MtrlOutByStatus”	L,n n = number of status dispositions 1. L,2 1. 51 (U1) MaterialStatus$_1$ 2. L,2 DestinationMap$_1$ 1. <CARRIERID$_{1,1}$> 2. L,m m = number of slots 1. <SLOTID$_1$> … m. <SLOTID$_m$> … n. L,2 1. 51 (U1) MaterialStatus$_n$ 2. L,2 DestinationMap$_n$ 1. <CARRIERID$_{n,1}$> 2. L,m m = number of slots 1. <SLOTID$_1$> … m. <SLOTID$_m$> If m = 0, the substrates may be placed by the equipment in any available slot.
“MtrlOutSpec”	L,p p = number of mapping source/destination pairs 1. L,2 1. L,2 SourceMap$_1$ 1. <CARRIERID$_{1,1}$> 2. L,m m = number of slots 1. <SLOTID$_1$> … m. <SLOTID$_m$> 2. L,2 DestinationMap$_1$ 1. <CARRIERID$_{1,2}$> 2. L,m m = number of slots 1. <SLOTID$_1$> … m. <SLOTID$_m$> … p. L,2 1. L,2 SourceMap$_p$ 1. <CARRIERID$_{p,1}$> 2. L,n n = number of slots 1. <SLOTID$_1$> … n. <SLOTID$_n$> 2. L,2 DestinationMap$_p$ 1. <CARRIERID$_{p,2}$> 2. L,n n = number of slots 1. <SLOTID$_1$> … n. <SLOTID$_n$>

Attribute Name	*Attribute Data Form: SECS-II Structure*
“PauseEvent”	L,n n = number of collection events 1. <CEID$_1$> … n. <CEID$_n$>
“ProcessingCtrlSpec”	L,p p = number of process jobs assigned to control job 1. L,3 1. <PRJOBID$_1$> 2. L,m ControlRule$_1$ 1. L,2 1. 20 (A) RuleName$_1$ 2. (Any format) RuleValue$_1$ … m. L,2 1. 20 (A) RuleName$_m$ 2. (Any format) RuleValue$_m$ 3. L,n OutputRule$_1$ 1. L,2 1. 51 (U1) MaterialStatus$_1$ 2. (Any format) OutputRuleValue$_1$ … n. L,2 1. 51 (U1) MaterialStatus$_n$ 2. (Any format) OutputRuleValue$_n$ … p. L,3 1. <PRJOBID$_p$> 2. L,m ControlRule$_p$ 1. L,2 1. 20 (A) RuleName$_1$ 2. (Any format) RuleValue$_1$ … m. L,2 1. 20 (A) RuleName$_m$ 2. (Any format) RuleValue$_m$ 3. L,n OutputRule$_p$ 1. L,2 1. 51 (U1) MaterialStatus$_1$ 2. (Any format) OutputRuleValue$_1$ … n. L,2 1. 51 (U1) MaterialStatus$_n$ 2. (Any format) OutputRuleValue$_n$ ControlRule 1) m = 0 indicates no ControlRule specified 2) RuleName and RuleValue are equipment specific OutputRule 1) n = 0 indicates no OutputRule specified 2) There is no defined OutputRuleValue. The content and format of OutputRuleValue is equipment dependent per SEMI E94 intent.

Attribute Name	*Attribute Data Form: SECS-II Structure*
“ProcessOrderMgmt”	51 (U1) ProcessOrderMgmt ProcessOrderMgmt is enumerated as follows: 1 = ARRIVAL 2 = OPTIMIZE 3 = LIST
“PRJobStatusList”	L,n n = number of process jobs defined in the control job 1. L,2 1. <PRJOBID$_1$> 2. <PRSTATE$_1$ >: n. L,2 1. <PRJOBID$_n$> 2. <PRSTATE$_n$>
“StartMethod”	11 (BOOLEAN) StartMethod StartMethod is as follows: TRUE – Auto FALSE – UserStart
“State”	51 (U1) State State is enumerated as follows: 0 = QUEUED 1 = SELECTED 2 = WAITINGFORSTART 3 = EXECUTING 4 = PAUSED 5 = COMPLETED

NOTICE: SEMI makes no warranties or representations as to the suitability of the standards set forth herein for any particular application. The determination of the suitability of the standard is solely the responsibility of the user. Users are cautioned to refer to manufacturer's instructions, product labels, product data sheets, and other relevant literature, respecting any materials or equipment mentioned herein. These standards are subject to change without notice.

By publication of this standard, Semiconductor Equipment and Materials International (SEMI) takes no position respecting the validity of any patent rights or copyrights asserted in connection with any items mentioned in this standard. Users of this standard are expressly advised that determination of any such patent rights or copyrights, and the risk of infringement of such rights are entirely their own responsibility.

SEMI E95-1101
SPECIFICATION FOR HUMAN INTERFACE FOR SEMICONDUCTOR MANUFACTURING EQUIPMENT

This specification was technically approved by the Global Information and Control Committee and is the direct responsibility of the North American Information and Control Committee. Current edition approved by the North American Regional Standards Committee on August 27, 2001. Initially available at www.semi.org September 2001; to be published November 2001. Originally published February 2000.

1 Purpose

1.1 This standard addresses the area of processing content with the direct intention of developing common software standards, so that problems involving operator training, operation specifications, and efficient development can be resolved more easily.

1.2 This standard is written to be “tool-neutral” without reference to, or reliance on, specific capabilities of platforms or operating systems. Neither is it intended that choices of software tools or detailed implementation strategies be dictated.

1.3 Note that all figures in this standard are schematic, are not drawn to scale, and unless otherwise specified, are not intended to provide implementation details about number of buttons, button sizes, panel sizes, etc.

2 Scope

2.1 This standard specification applies to manufacturing equipment used in the production of semiconductors.

2.2 This standard may be applicable to other areas such as the manufacture of flat panel displays, but specific application to these areas is outside the scope of this document.

2.3 This standard does not purport to address safety issues, if any, associated with its use. It is the responsibility of the users of this standard to establish appropriate safety and health practices and determine the applicability of regulatory limitations prior to use.

3 Referenced Standards

None.

4 Terminology

4.1 *condition* — a property of a displayed object or value (textual or numeric) that visually indicates, if applicable, whether the current state of an object or the current value violates the defined bounds of normal operational states or parameters, whether that violation is categorized as a minor exception (a caution) or a severe exception (an alarm), and provides no visual indication if no exception has occurred.

4.2 *display objects* — user interface elements displayed on the screen, such as function selection buttons, keyboard input buttons, graphics representing the equipment, etc.

4.3 *functional area* — a grouping of one or more views presenting information and control capabilities to the user.

4.4 *icon* — an icon (diagrammatic image) is a bitmap or other image used in GUI environments such as windowing systems to show different types of objects, improve operability, and help the user better understand the functionality underlying buttons.

4.5 *navigation model* — the navigation model determines how a user interacts with a system to access functionality and information.

4.6 *salience* — a salience is a solid (or textured), colored border shown around a display object to indicate an alarm, caution, or other condition, or to draw the user’s attention to the display object.

5 Requirements

5.1 Each of the following sections is designated with one of the following labels:

5.1.1 Sections designated:

Description	provide background information and set the context for subsequent specifications;
Mandatory	provide a specification of a requirement which shall both be present and implemented as specified;
Conditional	provide a specification of a requirement which shall be implemented as specified if such a feature is implemented on the tool;
Recommended	provide a recommended capability and implementation of that capability, but neither their presence nor their implementation is required.

5.2 *Basic Display Objects* Description

5.2.1 This section specifies the general appearance and behavior of basic display objects used throughout the interface, including buttons, saliences, and dialog boxes. It is intended that the use of other types of display objects (choose lists, data display and data entry fields, scroll-bars, etc.) is specifically allowed, and their use is at the discretion of the implementers.

5.2.1.1 There are two types of display objects; selectable and non-selectable. Some are selectable by the user to initiate or execute an action. Non-selectable graphics and user interface elements (such as pipes and text field labels, respectively) are read only, and no action is initiated or executed.

5.2.2 *Buttons*

5.2.2.1 *Button Size* Mandatory

5.2.2.1.1 When a touchscreen device is used, button sizes must be large enough to ensure selection on the targeted display.

5.2.2.2 *Button Dimensions* Recommended

5.2.2.2.1 It is recommended that buttons shall have a minimum dimension of approximately 1.5 cm on the shortest side. If a smaller size is used, the space between buttons must be increased to avoid selection errors. For installations where a keyboard and a mouse, light pen or other pointing device is available, button sizes in the navigation panel and the command panel may be made somewhat smaller (approximately 1–1.25 cm), and the size of the information panel increased proportionally.

5.2.2.3 *Button Behavior* Mandatory

5.2.2.3.1 One type of button behavior is momentary; that is, user selection of a button causes a brief display of the down (selected) state of the button, followed immediately by a display of the up (unselected) state. The other button behavior is two-state; the button remains in the down state after user selection. User re-selection of the button, and/or selection of another button, and/or selection of another display object restores the display of the up state. In some cases, the software will control the display of the down state or restore the up state, without direct user interaction.

5.2.2.3.2 For 2-D buttons, the down state shall be indicated by hatching, cross-hatching, or otherwise texturing the button in such a manner that does not obscure the button label.

5.2.2.4 *Button Text* Mandatory

5.2.2.4.1 Text for all button labels shall have the first letter of words capitalized unless it is an article or preposition not occurring at the beginning or end of the label, or unless the word's conventional usage is not capitalized. Button labels that are all capital letters are harder to read than mixed case labels. Additionally, text in all capitals appears larger, and the user may attach more importance to the button than necessary simply because the label is visually distracting.

5.2.3 *Saliences* Conditional

5.2.3.1 Saliences, colored, textured, or both shall be displayed around buttons and other display objects to indicate their condition, which may include caution, alarm, user attention required or requested, processing, unfinished task notification, and other conditions.

5.2.3.2 A salience is displayed to draw the user's attention to a display object when its condition is *not* normal or OK (in this case, the absence of a displayed salience shall indicate a normal or OK condition), or when the salience provides information that benefits the user in the performance of tasks or the monitoring of equipment functions and operations.

5.2.3.3 A salience shall not hide the display object it surrounds. Saliences shall not be used to indicate the state (open, closed, on, off, etc.) of display objects.

5.2.3.4 On color displays, alarm saliences shall appear bright red, caution saliences shall appear bright yellow, and processing and unfinished task saliences shall appear medium blue. User attention required or requested saliences, (for example, "Ready to Load," or "Ready to Unload") shall appear medium green.

5.2.4 *Dialog Boxes* Description

5.2.4.1 Dialog boxes are secondary windows used to display supplemental information, solicit information from the user, or report errors.

5.2.4.2 Dialog boxes are used to provide additional information to the user; to display detailed information not shown on the information panel for controlling the system, and to display detailed information for monitoring system operation.

5.2.5 *Dialog Boxes* Mandatory

5.2.5.1 Dialog boxes (which are always temporary) are displayed in response to some action initiated by the user. When displayed, a dialog box shall overlay a portion of the information panel, and shall not obscure the title panel. If invoked by user selection of a display object on the information or command panels, all the display objects on those two panels shall be disabled until the dialog box is dismissed. The title and navigation panels remain enabled. At the explicit request of the user, the dialog box is dismissed, and the underlying information is refreshed.

5.2.5.2 Dialog boxes contain a title bar at the top, a display (free use) area, and one or more dialog box window control buttons arranged horizontally at the bottom. The title bar text reflects the command or the nature of the event that invoked the dialog box. Dialog box window control buttons are centered on the width of the dialog box, with any other buttons (Apply, Log-out, etc.) right-aligned and visually separated from the window control buttons. If the underlying operating system will not allow this alignment, then it is allowed that other alignments may be used, but only if the alignment is consistent across all dialog boxes.

5.2.5.2.1 User selection of a dialog box control button controls the dismissal of the dialog box and, when applicable, controls whether the user accepts or rejects information or choices displayed, or desires no action be performed.

5.2.5.2.1.1 An “OK” dialog box control button, when selected by the user, indicates acceptance of any choices or user inputs made, if any, and causes the dismissal of the dialog box. If no choices or user inputs were made, selecting this button indicates acceptance of any default values displayed. If user choices or inputs are required, this button shall be disabled until the choices or inputs are made.

5.2.5.2.1.2 A “Cancel” dialog box control button, when selected by the user, indicates no action should be taken, causes the dismissal of the dialog box, and returns the user to the state that existed prior to the invocation of the dialog box.

5.2.5.2.1.3 A “Yes” dialog box control button is displayed when the dialog box message is in the form of a question. User selection indicates a positive response to the question asked and causes the dismissal of the dialog box.

5.2.5.2.1.4 A “No” dialog box control button is displayed when the dialog box message is in the form of a question. User selection indicates no action should be taken, causes the dismissal of the dialog box, and returns the user to the state that existed prior to the invocation of the dialog box.

5.2.5.2.1.5 A “Close” dialog box control button is displayed (often as the only dialog box control button) when the dialog box message contains only information and does not require the user to make or accept choices, and shall be used instead of an “OK” dialog box control button in this case. The “Close” dialog box control button shall also be used instead of a “Cancel” dialog box control button when the user cannot be returned to the state that existed prior to the invocation of the dialog box. User selection indicates no action should be taken and causes the dismissal of the dialog box.

5.2.5.2.1.6 An “Apply” dialog box control button, when selected by the user, indicates acceptance of any choices or user inputs made, if any, but does not dismiss the dialog box. This button shall be disabled until one or more user choices or inputs are made. After user selection of this button, it shall be disabled until additional choices or user inputs are made, if any.

5.2.5.3 Dialog boxes may not be resized or moved, but may display a button equivalent to the Cancel button in the title bar.

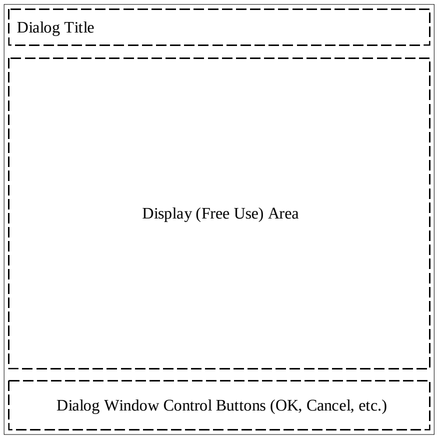

Figure 1
Dialog Box

5.2.5.4 Dialog boxes are classified into the following three types:

- Information dialog box
- Data input/selection dialog box
- Message dialog box

5.2.5.5 To be compliant with this specification, at least one of these dialog box types shall be supported.

5.2.6 *Information Dialog Box* Conditional

5.2.6.1 This dialog box type is used to provide additional information to the user about some display object or topic addressed by the information panel. User selection of a display object on the information panel invokes an information dialog box, if appropriate for the display object selected. The window control button to dismiss the dialog box shall be the Close button. Use of an OK button in this case is not allowed.

5.2.6.2 Implementation of this dialog box type shall be conditional on the equipment having the capability of providing the required information.

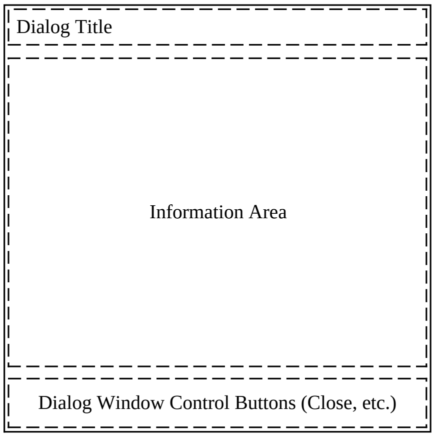

Figure 2
Information Dialog Box

5.2.7 *Input/Selection Dialog Box* Conditional

5.2.7.1 This dialog box type is used to request data input or selection from the user. If no keyboard or keypad is available, and the user must input characters, an on-screen representation ("mimic") of one or both shall be displayed as part of the dialog box. The window control buttons are the OK and Cancel buttons.

5.2.7.2 Implementation of this dialog box type shall be conditional on the presence of display objects that allow user input or selection.

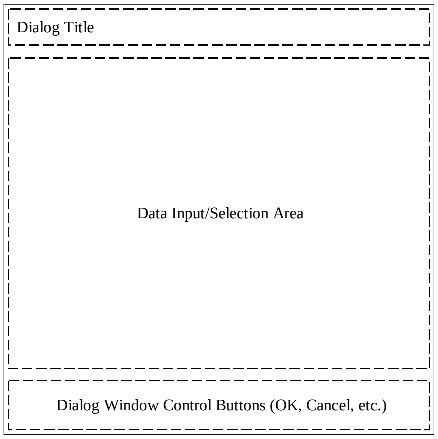
Dialog Title
Data Input/Selection Area
Dialog Window Control Buttons (OK, Cancel, etc.)

Figure 3
Data Input/Selection Dialog Box

5.2.8 *Message Dialog Box* Conditional

5.2.8.1 This dialog box type is used to provide a message to the user (including the reporting of errors) or to request confirmation of a user initiated action. The message text is located to the right of the icon. It is recommended that icons from the underlying operating system be used to represent the message type as follows:

- Information in the form of a simple message
- Progress, informing the user of an ongoing process
- Attention, alerting the user of possible danger, or inability to execute a command, or requesting confirmation (either as a statement or as a question)
- Error, informing the user of danger or inability to execute a command (if effect is severe)

5.2.8.2 Note that most style guides no longer recommend the use of a question mark icon when the message is phrased as a question, as its meaning could be ambiguous in some cases. The attention icon should be used instead.

5.2.8.3 The first two message types use the Close window control button. The second two use OK and Cancel, or Yes and No (and sometimes Cancel) if the message is phrased as a question. It is recommended that buttons or other display objects that would cause an error message be disabled in those circumstances.

5.2.8.4 Implementation of this dialog box type shall be conditional on the equipment having the capability of providing the required information.

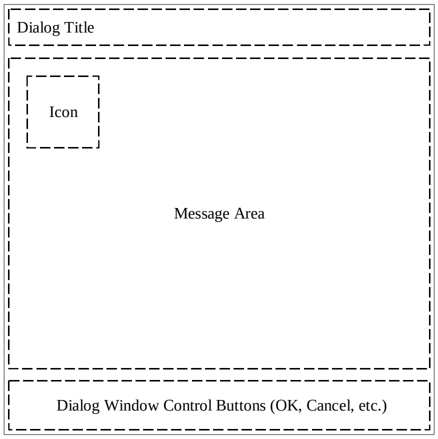

Figure 4
Message Dialog Box

5.3 *Basic Network Navigation Model* Mandatory

5.3.1 This standard specifies a simple navigation model designed specifically to minimize the number of actions and the amount of time required of the user.

5.3.2 The basic network navigation model is capable of displaying a number of views within a single level of hierarchy. The user does not have to traverse up and down menu or view “trees” when exercising control and monitoring tasks. As shown in the figure below, the basic navigation model is a network, supporting horizontal (lateral) transfer, at any time, between any of the functional areas in the interface. The basic network navigation model does not support more than one view in any functional area. Expansion of detail for views is typically implemented using dialog boxes.

5.3.3 Note that the diagram is schematic; it shows that user selection of a functional area shall display its associated information panel. Only one information panel shall be displayed at any time.

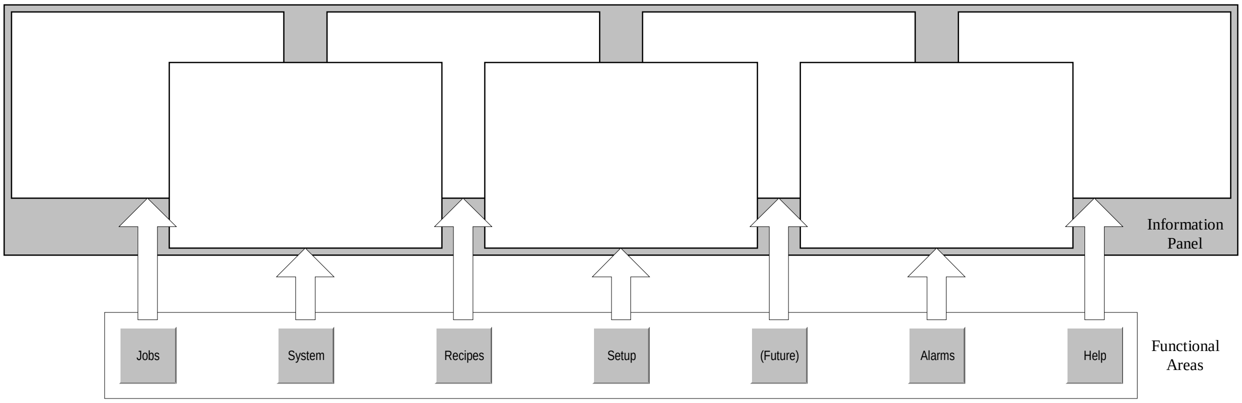

Figure 5
Basic Network Navigational Model

5.3.4 *Network Navigation Model with Sub-navigation* Conditional

5.3.4.1 While maintaining the same basic structure, this navigation model supports multiple views within functional areas on the network. If any functional area has more than one view, all functional areas in the interface shall consistently use one of the two view sub-navigation methods described below. Lateral transfer between any of the views within a functional area is supported either by providing a single row of tabs which may be selected to change views (Figure 6), or by providing view sub-navigation buttons (Figure 7) in a separate screen area dedicated solely to sub-navigation. Expansion of detail for each of these views is typically implemented using dialog boxes.

5.3.4.2 The grouping of tasks within functional areas reflects the natural flow of information, events, and tasks in a way that is familiar to the user and that directly supports the attainment of successful process and equipment performance goals. Functional areas are user task oriented, collecting together logically related monitoring and control functions, reducing the need to navigate between views.

5.3.4.3 Note that the diagrams are schematic; they show that user selection of a functional area shall display its associated information panel. Only one information panel shall be displayed at any time. Similarly, user selection of a tab or sub-navigation button displays only one of the views available at any time.