Table 96 Assembly-EPD#4 Object Instance Network Variables

Assembly-EPD#4 Profile ID = 180.97, Instance ID = 01				
Sequence Number	*Name*	*Storage Class*	*SDM Tag*	*Standard NV or CP Data Type*
1	Data	File	A1	LonMark file transfer NVs and messaging interface.

Table 97 Assembly-EPD#4 Object Instance Network Services

Assembly-EPD#4 Profile ID = 180.97, Instance ID = 01				
Service Request Code	*Service Name*	*SDM Tag*	*Request Parameters*	*Result Parameters*
--	No Additional Services Defined			

SEMI E54.17-0705
SPECIFICATION OF SENSOR/ACTUATOR NETWORK FOR A-LINK

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 May 20, 2005. It was available at www.semi.org in June 2005 and on CD-ROM in July 2005.

1 Purpose

1.1 This document defines a communication specification based on the A-LINK protocol to enable communications between intelligent devices on a sensor/actuator network (SAN) that operate according to SEMI E54 device models on semiconductor manufacturing equipment.

1.2 This document gives interoperability with SEMI E54 common/specific device model based Sensor/Actuator devices.

2 Scope

2.1 This document specifies how Sensor / Actuator / Controller devices interoperate on the network specific for the A-LINK Public specification referenced in the §4, as a part of equipment's control system.

2.2 This document defines relation with main part of SEMI E54 including Common Device Model (CDM) and existing Specific Device Models (SDMs). This document is to be used with the CDM and one or more SDMs, as well as the A-LINK Public Specification.

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 document specifies a semiconductor equipment SAN based only A-LINK and is a companion document to the A-LINK specification; therefore a complete specification of this standard includes the A-LINK specifications. There are other semiconductor SAN communication options. The specifications for these options are not included here.

3.2 This standard specifies enhancements that provide additional capability over and above those currently required by A-LINK. In order to avoid document inconsistency problem, information in the A-LINK specification that relates to this standard is not repeated in this document. This document is limited to describing enhancements or limitations to the A-LINK specification that are imposed by this standard.

3.3 A complete specification of the conformance testing procedure shall include the A-LINK protocol conformance testing specification. Conformance testing shall also include enhancements and limitations to the A-LINK specification required by this standard.

4 Referenced Standards and Documents

4.1 *SEMI Standards*

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

SEMI E54.1 — Standard for Sensor/Actuator Network Common Device Model

SEMI E54.3 — Specification for Sensor/Actuator Network Specific Device Model for Mass Flow Device

SEMI E54.10 — Specification for Sensor/Actuator Network Specific Device Model for an In-situ Particle Monitor Device

SEMI E54.11 — Specific Device Model for Endpoint Devices

4.2 *OSI Standard*

ISO 7498 OSI — Basic Reference Model for Open Systems Interconnection

4.3 *A-LINK Documents*[1]

A-LINK Public Specification ver. 1.2 — Basic Reference Model for Open Systems Interconnection

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

5 Terminology

5.1 *Abbreviations and Acronyms*

5.1.1 *AUF* — A-LINK User Forum

5.1.2 *CDM* — Common Device Model

5.1.3 *NCS* — Network Communication Standard

5.1.4 *OSI* — Basic Reference Model for Open Systems Interconnection (ISO 7498)

5.1.5 *PHY* — Physical Layer

5.1.6 *SAN* — Sensor/Actuator Network

5.1.7 *SDM* — Specific Device Model

5.1.8 *UI* — User Interface

5.2 This section includes the terms defined in SEMI E54.1 Sensor/Actuator Network Common Device Model.

5.2.1 *Attribute*

5.2.2 *Behavior*

5.2.3 *Byte*

5.2.4 *Class*

5.2.5 *Common Device Model*

5.2.6 *Device*

5.2.7 *Device Manager (DM) Object*

5.2.8 *Device Model*

5.2.9 *Instance*

5.2.10 *Network Communication Standard*

5.2.11 *Object*

5.2.12 *Sensor, Actuator and Controller (SAC) Object*

5.2.13 *Service*

5.2.14 *Specific Device Model*

5.2.15 *state diagram*

5.3 *Terminology Defined in A-LINK*

5.3.1 *AUF* — A-LINK User Forum.[1] A kind of consortium for A-LINK users to recommend improvement to A-LINK trade organization[2], share A-LINK applications, provide A-LINK compliance test and etc.

5.3.2 *AN MS* — a station that accesses and exchanges data by polling its assigned Networked-Slave stations.

5.3.3 *AN SS* — a station that is managed and accessed by AN MS.

---

1 http://www.a-linkuf.com

2 http://www.algosystem.co.jp

5.3.4 *Device Profile* — a kind of electric table to provide the characteristic features of a device including configuration and capabilities.

5.3.5 *Service Access Element* — an addressable location in a device for the directing of service requests.

5.3.6 *Data Exchange* — a capability to communicate such data for a device as raw/processed sensed data and variable settings.

5.3.7 *Directive* — a capability to instruct such basic functions to a device as reset or abort.

5.3.8 *DP Facility* — a capability to manage diagnostic matters.

5.3.9 *Unite Inventory* — a capability to realize and communicate the Device Profile.

5.3.10 *Unite States* — a capability to communicate state of a device.

6 Communication Protocol High Level Structure

6.1 In a typical remote I/O configuration, single master architectures are used to optimize response times. In complicated applications, multi-master architectures are also possible. A-LINK uses the polling principle for communication.

6.1.1 Message transfer is organized in cycles. A message cycle mainly consists of a request-frame followed by a corresponding acknowledge/response-frame of the addressed station. An exception to this is the global-control function for synchronization and coordination of several remote I/O stations.

6.1.2 A brief description of the A-LINK protocol as it relates to the ISO 7498 OSI model follows in the sections below. For protocol efficiency, A-LINK does not define layers 3 to 7. However, since the OSI model specifies Layer 7 as the interface between the Application Process and the communication stack, it is appropriate to discuss several aspects of the A-LINK standard at this level.

NOTE 1: The information contained in this section is for reference only. It in no way represents specifications for A-LINK. See related documentation for these specifications.

6.2 *Physical Layer* — Layer 1

6.2.1 The bottom Physical Layer is established by RS-485. See the A-LINK standard for more information for detail.

6.3 *Data Link Layer* — Layer 2

6.3.1 *Data Transfer*

6.3.1.1 The Data Link Layer provides the functions for sending and receiving data over the network. Data Elements are packaged, delivered, and checked. Acknowledgements, responses, retries, and timeouts are used to guard against Line Protocol Errors (e.g., frame, overrun, and parity) and Transmission Protocol Errors (e.g., start and end delimiters, frame check, frame length, and response times).

6.4 *Network Layer* — Layer 3

6.4.1 There is no distinct network layer.

6.5 *Transport Layer* — Layer 4

6.5.1 There is no distinct transport layer. Such functions as disassembling of message into transport unit and reassembling for transportation are implemented in the Application Layer.

6.6 *Session Layer* — Layer 5

6.6.1 There is no distinct session layer.

6.7 *Presentation Layer* — Layer 6

6.7.1 There is no distinct presentation layer. Such functions as data representation conformance are implemented in the Application Layer.

6.8 *Application Layer* — Layer 7

6.8.1 The Device shall comply with A-LINK application layer specification for defining and addressing objects including functions deferred in lower layers as described above.

6.8.2 *User Interface (UI)*

6.8.2.1 The UI provides the user with access to functionality of the A-LINK protocol as a part of application layer.

7 Required Object Types

7.1 This section describes a general mapping of the SEMI SAN Object Model to the A-LINK environment. Component definitions are clarified and the mapping of Attributes, Services, and Behaviors are specified.

7.2 *Object Model*

7.2.1 The Object Model defined in the CDM is represented in the A-LINK NCS. Especially the DM, SAC, and abstract or fundamental application objects are mapped.

7.2.2 The Application Objects associated with the SDM standards are mapped in A-LINK User Forum Public Specification documents. §9 specifies the mapping of SDM Objects in A-LINK.

7.3 *Component Mapping Summary*

7.3.1 Table 1 provides a summary of the components of the CDM object model as they relate to the components of A-LINK.

7.4 *Objects*

7.4.1 The required objects of the CDM are identified here. Additional objects that are contained in the SDM are given identifiers in the Device Profile. §9 specifies additional mapping information.

7.4.2 Table 1 lists the Object Identifiers specified for use in protocol messages.

Table 1 Object Identifiers

Object Name	*A-LINK Class/Object ID*	*CDM Object ID Tag*	*CDM Attribute/Service ID Tag Prefix*
Device Manager	1	DmI0	Dm
SAC	2	SacI0	Sac
Assembly	3	AsmIn	Asm
Local Link	4	LnkIn	Lnk
Sensor–AI	9	SenIn	Sai
Sensor–EI	10	SenIn	Sei
Sensor–BI	11	SenIn	Sbi
Sensor–BI–TH	12	SenIn	Sbith
Actuator–AO	17	ActIn	Aao
Actuator–EO	18	ActIn	Aeo
Actuator–BO	19	ActIn	Abo
Controller	24	CntIn	C
Application Objects	>31	-	-

7.5 *Attributes*

7.5.1 All attributes are accessible via Get_Attribute and Set_Attribute services defined in the sections below. Attributes are also accessible via different A-LINK peculiar instructions which are additionally mapped in this document based on attribute type.

7.5.1.1 The attributes of the DM object consists of distinctive information, states, and setups. Distinctive information attributes are retrieved with the Unite Inventory. Inquiring state attributes are provided by getting status data. Setup attributes are written by setting configuration data.

7.5.1.2 The attributes of Application objects are divided into two types: Input/Output, settings, status and Configuration. Input/Output attributes are acquired from or given through Data Exchange capability of A-LINK. Configurable attributes can be retrieved with Unite Inventory capability.

7.5.1.3 See Table 2 for a list of DM attributes and their related alternative access instructions.

7.5.1.4 *A-LINK Settings*

7.5.1.4.1 Attributes related to general settings are accessed through Unite Inventory communication.

7.5.1.4.2 Structure of the Unite Inventory for a given SDM is beyond the scope of this document. The A-LINK Trade Organization is responsible for the management of this information.

7.5.1.5 *A-LINK States and Diagnostics*

7.5.1.5.1 Attributes related to device state are categorized and handled as Unite State information. For example two attributes of the DM object listed in Table 2 that are identified with an alternative access instruction of Unite States are mapped into the A-LINK Unite States as specified in this section. See the A-LINK standard for a description of the Unite States.

7.5.1.5.2 These two attributes are mapped into the Unite State data structure. Additional diagnostic data may be included with DP Facility as specified by A-LINK.

7.5.2 *A-LINK I/O Data Exchange*

7.5.2.1 Input/Output attributes of the Application objects are communicated using the I/O Data Exchange instruction of A-LINK. This instruction is described in the A-LINK standard. A list of which attributes are accessible with this instruction is included in the A-LINK Device Profile for a given device type.

7.5.3 *A-LINK Device Configuration*

7.5.3.1 Configuration attributes of the DM object and Application objects are communicated using the Unite Inventory Service of A-LINK. This service is described in the A-LINK standard. A list of which Application object attributes are accessible with this instruction is included in the A-LINK Device Profile for a given device type.

7.5.3.2 *Attribute Identifiers*

7.5.3.2.1 Every class object specified in the CDM uses attribute identifier tags to identify its attributes. The tags are formed with alphabetic part of its object identifier tag, a character 'A' for impressing attribute and a numerical identifier of the attribute. The numerical identifier is assigned to the Attribute ID used in the A-LINK NCS for classes DM, SAC, Assembly, Local Link, AE, S, A and C, as described in following tables: i.e. from Table 2 through Table 8.

7.5.3.2.2 Except Sensor-Binary Input Threshold (SBITH) class, the Attribute ID used in the A-LINK for derived classes of Sensor and Actuator classes is assigned with the same manner as above.

7.5.3.2.3 For Sensor-Binary Input Threshold (SBITH) class, the Attribute ID used in the A-LINK starts from 128 to prevent duplication with the numerical identifiers in CDM assigned and reserved to is parent class, SBI.

7.5.3.2.4 The A-LINK attribute ID is used to identify attributes for access via A-LINK message requests, which are explained in later sections.

7.5.4 *A-LINK Attribute Mappings for CDM*

7.5.4.1 *Device Manager (DM) Object* — The DM object is the device component responsible for managing and consolidating the device operation as defined in the CDM standard. Attribute values including DM state are also as same as defined in the CDM standard. The presentation of object instance attributes to the A-LINK network shall be as indicated in Table 2.

Table 2 DM Object Attribute Identifiers for A-LINK

Attribute ID		*Attribute Name*	*Alternative Access*
CDM	*A-LINK*		
DmA1	1	Device Type	Unite Inventory
DmA2	2	Standard Revision Level	Unite Inventory

Attribute ID		Attribute Name	Alternative Access
CDM	*A-LINK*		
DmA3	3	Device Manufacturer Identifier	Unite Inventory
DmA4	4	Manufacturer Model Number	Unite Inventory
DmA5	5	Software or Firmware Revision Level	Unite Inventory
DmA6	6	Hardware Revision Level	Unite Inventory
DmA7	7	Serial Number	Unite Inventory
DmA8	8	Device Configuration	Unite Inventory
DmA9	9	Device Status	Unite States
DmA10	10	Reporting Mode	Unite Inventory
DmA11	11	Exception Status Report Interval	Unite Inventory
DmA12	12	Exception Status	Unite States
DmA13	13	Exception Detail Alarm	Unite Inventory
DmA14	14	Exception Detail Warning	Unite Inventory
DmA15	15	Visual Indicator	Unite Inventory
DmA16	16	Alarm Enable	Unite Inventory
DmA17	17	Warning Enable	Unite Inventory
DmA18	18	Exception Detail Type	Unite Inventory
DmA19	19	Exception Detail Alarm Queue	Unite Inventory
DmA20	20	Exception Detail Warning Queue	Unite Inventory
DmA21	21	Date and Time	Unite Inventory
DmA22	22	Date and Time Type	Unite Inventory

7.5.4.2 *Sensor, Actuator, Controller (SAC) Object* — The SAC object is the device component responsible for coordinating the interaction of the device with the sensory/actuation/control environment as defined in the CDM standard. Attribute values including SAC state are also as same as defined in the CDM standard. The presentation of object instance attributes to the A-LINK network shall be as indicated in Table 3.

Table 3 SAC Object Attribute Identifiers for A-LINK

Attribute ID		Attribute Name	Alternative Access
CDM	*A-LINK*		
SacA1	1	Last Calibration Date	DP Facility
SacA2	2	Next Calibration Date	DP Facility
SacA3	3	Expiration Timer	DP Facility
SacA4	4	Expiration Warning Enable	DP Facility
SacA5	5	Run Hours	DP Facility

7.5.4.3 *Assembly Object (Asm)* — The Assembly (Asm) object instances may be used to provide for grouping more than one attribute from one or more object instances as defined in the CDM standard. Attribute values are also as same as defined in the CDM standard. The presentation of object instance attributes to the A-LINK network shall be as indicated in Table 4.

Table 4 Assembly Object Attribute Identifiers for A-LINK

Attribute ID		Attribute Name	Alternative Access
CDM	*A-LINK*		
AsmA1	1	Data	Data Exchange

7.5.4.4 *Local Link Object (Lnk)* — The Local Link (Lnk) object instances may be used to 'link" an attribute of one object instance to an attribute of another object instance as defined in the CDM standard. Attribute values are also as same as defined in the CDM standard. The presentation of object instance attributes to the A-LINK network shall be as indicated in Table 4.

Table 5 Local Link Object Attribute Identifiers for A-LINK

Attribute ID		*Attribute Name*	*Alternative Access*
CDM	*A-LINK*		
LnkA1	1	Source Object Class	Unite Inventory
LnkA2	2	Source Object Instance	Unite Inventory
LnkA3	3	Source Object Attribute	Unite Inventory
LnkA4	4	Destination Object Class	Unite Inventory
LnkA5	5	Destination Object Instance	Unite Inventory
LnkA6	6	Destination Object Attribute	Unite Inventory
LnkA7	7	Commit	Unite Inventory

7.5.4.5 *Active Element (AE) Class* — The AE class is an abstract class generic to any device component as defined in the CDM standard. Attribute values including AE state are also as same as defined in the CDM standard. The presentation of object instance attributes to the A-LINK network shall be as indicated in

7.5.4.6 These attributes are inherited by any sensor, actuator and controller objects, and such further derived objects as Sensor-AI and Actuator-BO objects.

Table 6 AE Class Attribute Identifiers for A-LINK

Attribute ID		*Attribute Name*	*Alternative Access*
CDM	*A-LINK*		
nA1	1	Name	Unite Inventory
nA2	2	Status	Unite States
nA3	3	AlarmEnable	Unite Inventory
nA4	4	WarningEnable	Unite Inventory

[#1] Prefix n in CDM attribute ID represents one of Sai, Sei, Sbi, Sbith, Aao, Aeo, Abo and C.

7.5.4.7 *Sensor (S) Class* — The S class is an abstract class generic to any sensors on device component as defined in the CDM standard. Attribute values are also as same as defined in the CDM standard. The presentation of object instance attributes to the A-LINK network shall be as indicated in Table 7. These attributes are inherited by any sensor objects as Sensor-AI object. They also inherit AE class attributes. Attribute identifiers on A-LINK for such direct and indirect inheritor classes of S class as Sensor-AI are assigned as ¶7.5.3.2 and Data Exchange communication could give alternative access for the attributes.

Table 7 S Class Attribute Identifiers for A-LINK

Attribute ID		*Attribute Name*	*Alternative Access*
CDM	*A-LINK*		
nA16	16	Value	Data Exchange
nA17	17	ReportInhibitTimer	Data Exchange
nA18	18	EnableReportRate	Data Exchange
nA19	19	ReportRate	Data Exchange

[#1] Prefix n in CDM attribute ID represents one of Sai, Sei, Sbi or Sbith.

7.5.4.8 *Actuator (A) Class* — The A class is an abstract class generic to any actuators on device component as defined in the CDM standard. Attribute values are also as same as defined in the CDM standard. The presentation of object instance attributes to the A-LINK network shall be as indicated in Table 8. These attributes are inherited by any sensor objects as Actuator-AO object. They also inherit AE class attributes. Attribute identifiers on A-LINK for inheritor classes Sensor-AO are assigned as ¶7.5.3.2 and Data Exchange communication could give alternative access for the attributes.

Table 8 A Class Attribute Identifiers for A-LINK

Attribute ID		*Attribute Name*	*Alternative Access*
CDM	*A-LINK*		
nA16	16	Setting	Data Exchange
nA17	17	SafeState	Data Exchange
nA18	18	WatchRate	Data Exchange
nA19	19	WatchDog	Data Exchange

[#1] Prefix n in CDM attribute ID represents one of Aao, Aeo or Abo.

7.5.4.9 *Controller (C) Object* — The C object contains structure and behavior common to all controller element instances on device component as defined in the CDM standard. Attribute values are also as same as defined in the CDM standard. The presentation of object instance attributes to the A-LINK network shall be as indicated in Table 9. It inherits AE class attributes.

Table 9 C Class Attribute Identifiers for A-LINK

Attribute ID		*Attribute Name*	*Alternative Access*
CDM	*A-LINK*		
CA16	16	Setpoint	Data Exchange
CA17	17	ProcessVariable	Data Exchange
CA18	18	ControlVariable	Data Exchange
CA19	19	DataType	Data Exchange
CA20	20	DataUnits	Data Exchange
CA21	21	AlarmSettleTime	Data Exchange
CA22	22	AlarmErrorBand	Data Exchange
CA24	24	WarningSettleTime	Data Exchange
CA25	25	WarningErrorBand	Data Exchange

7.6 *Services*

7.6.1 A-LINK specifies standard mechanisms for the communication of data over the network. These mechanisms are used to communicate attributes specified in the device on the A-LINK. Also they are used to request instructions specific for component in the device on the A-LINK.

7.6.2 *Service Requests and Response*

7.6.2.1 The A-LINK attribute communications and instructions require specific definitions. They are described in the following sections.

7.6.2.1.1 *Service Request and Response Protocol*

7.6.2.1.1.1 All service request messages, except Get_Attribute and Set_Attribute, are sent to a device using Directive service functions of A-LINK. The responses to these message requests are specified by A-LINK.

7.6.2.1.1.2 The Service Request message is formatted, as defined by A-LINK, with the following information:

Location := Object ID

7.6.2.1.1.3 This service request message has no other information requested with the message. Response of the request has following information:

Location := Object ID

Status := Response Information

NOTE 2: The Location information for the response message is optional (conditional) information. Representation of the Status information is implementation specific. The Response Information is dependent on class and/or device. However if the first item of the response information, i.e. Response Code, is zero, it always means successful. Additional data may follow.

7.6.2.1.2 *Set_Attribute Protocol*

7.6.2.1.2.1 The Service Request message for the Set-Attribute is sent to a device using writing capability for following service functions of A-LINK: Unite Inventory service, Data Exchange service or DP Facility service. Using service function is dependent on attribute and object class as well. The dependency is given by Object Service Identifier tables in following subsections of ¶7.6.

7.6.2.1.2.2 The service request message with Unite Inventory service function of A-LINK has a couple of information items. Information items for the request and its response are given below respectively:

(Request)
Location := Object ID :optional
Data Name ID := Attribute ID
Data Value := Attribute Value

(Response)
Location := Object ID ;optional
Data Name ID := Attribute ID
Data Value := Attribute Value
Status := Response Information

NOTE 3: The items of Data Name ID and Data Value for the response message are optional (conditional) information. Representation of the Status information is implementation specific. The Response Information is dependent on attribute and/or device. However if the first item of the response information, i.e. Response Code, is zero, it always means successful. Additional data may follow.

7.6.2.1.2.3 The service request message with Data Exchange service function of A-LINK has three information items. Information items for the request and its response are given below respectively:

(Request)
Location := Object ID
Data Name ID := Attribute ID
Data Value := Attribute Value

(Response)
Location := Object ID
Data Name ID := Attribute ID
Data Value := Attribute Value
Status := Response Information

NOTE 4: The items of Location, Data Name ID and Data Value for the response message are optional (conditional) information. Representation of the Status information is implementation specific. The Response Iinformation is dependent on attribute and/or class.

7.6.2.1.2.4 The service request message with DP Facility service function of A-LINK has a couple of information items. Information items for the request and its response are given below respectively:

(Request)
Information Name ID := Attribute ID
Information Value := Attribute Value

(Response)
Information Name ID := Attribute ID
Information Value := Attribute Value
Status := Response Information

NOTE 5: The items of Information Name ID and Information Value for the response message are optional (conditional) information. Representation of the Status information is implementation specific. The Response Information is dependent on attribute and/or device. However if the first item of the response information, i.e. Response Code, is zero, it always means successful. Additional data may follow.

7.6.2.1.3 *Get_Attribute Protocol*

7.6.2.1.3.1 The Service Request message for the Get-Attribute is sent to a device using reading capability for following service functions of A-LINK: Unite Inventory service, Unite State service, Data Exchange service or DP Facility service. Using service function is dependent on attribute and object class as well. The dependency is given by Object Service Identifier tables in following subsections of ¶7.6.

7.6.2.1.3.2 The service request message with Unite Inventory service function of A-LINK has an information item. Information items for the request and its response are given below respectively:

(Request)
Location := Object ID :optional
Data Name ID := Attribute ID

(Response)
Location := Object ID :optional
Data Name ID := Attribute ID
Data Value := Attribute Value
Status := Response Information

NOTE 6: The item of Data Name ID for the response message is optional (conditional) information. Representation of the Status information is implementation specific. The Response Information is dependent on attribute and/or device. However if the first item of the response information, i.e. Response Code, is zero, it always means successful. Additional data may follow.

7.6.2.1.3.3 The service request message with Unite State service function of A-LINK has an information item. Information items for the request and its response are given below respectively:

(Request)
Top State Name ID := Object ID

(Response)
Top State Name ID := Object ID
Data Value := Attribute Value
Status := Response Information

NOTE 7: The item of Top State Name ID for the response message is optional (conditional) information. Representation of the Status information is implementation specific. The Response Information is dependent on attribute and/or device. However if the first item of the response information, i.e. Response Code, is zero, it always means successful. Additional data may follow.

7.6.2.1.3.4 The service request message with Data Exchange service function of A-LINK has a couple of information items. Information items for the request and its response are given below respectively:

(Request)
Location := Object ID
Data Name ID := Attribute ID

(Response)
Location := Object ID
Data Name ID := Attribute ID
Data Value := Attribute Value
Status := Response Information

NOTE 8: The items of Location and Data Name ID for the response message are optional (conditional) information. Representation of the Status information is implementation specific. The Response Information is dependent on attribute and/or class. However if the first item of the response information, i.e. Response Code, is zero, it always means successful. Additional data may follow.

7.6.2.1.3.5 The service request message with DP Facility service function of A-LINK has an information item. Information items for the request and its response are given below respectively:

(Request)
Information Name ID := Attribute ID

(Response)
Information Name ID := Attribute ID

Information Value := Attribute Value
Status := Response Information

NOTE 9: The item of Information Name ID for the response message is optional (conditional) information. Representation of the Status information is implementation specific. The Response Information is dependent on attribute and/or device. However if the first item of the response information, i.e. Response Code, is zero, it always means successful. Additional data may follow.

7.6.2.2 *Service Identifiers* — The A-LINK services are mapped for CDM as following tables.

7.6.2.2.1 *Device Manager (DM) Object* — Services of the DM object are mapped as shown in Table 10.

Table 10 DM Object Service Identifiers for A-LINK

Service ID		*Service Name*	*Alternative Instruction*
CDM	*A-LINK*		
DmS1	1	Reset	Directive
DmS2	2	Abort	Directive
DmS3	3	Recover	Directive
DmS4	4	GetAttribute	Unite Inventory / Unite States
DmS5	5	SetAttribute	Unite Inventory
DmS6	6	Execute	Directive
DmS7	7	PerformDiagnostics	Directive
DmS8	8	PublishAttribute	Directive
DmS9	9	Lock	Directive
DmS10	10	Unlock	Directive
DmS11	11	Get Exception Queue	Directive
DmA12	12	Clear Exception Queue	Directive

7.6.2.2.2 *Sensor Actuator Controller (SAC) Object* — Services of the SAC object are mapped as shown in Table 11.

Table 11 SAC Object Service Identifiers for A-LINK

Service ID		*Service Name*	*Alternative Instruction*
CDM	*A-LINK*		
SacS1	1	Reset	Directive
SacS2	2	Abort	Directive
SacS3	3	Recover	Directive
SacS4	4	GetAttribute	Unite Inventory / Unite States
SacS5	5	SetAttribute	Unite Inventory
SacS6	6	Operate	Directive
SacS7	7	Restore Default	Directive
SacS8	8	Publish Attribute	Directive

7.6.2.2.3 *Active Element (AE) Object* — Services of the AE object are mapped as shown in Table 12.

7.6.2.2.3.1 Because AE class is inherited to any sensor, actuator or controller objects such as Sensor-AI and Actuator-BO, these services are equipped by such objects.

Table 12 AE Object Service Identifiers for A-LINK

Service ID		*Service Name*	*Alternative Instruction*
CDM	*A-LINK*		
nS1	1	Reset	Directive
nS2	2	Abort	Directive
nS3	3	Recover	Directive
nS4	4	Operate	Directive
nS5	5	GetAttribute	Data Exchange

Service ID		Service Name	Alternative Instruction
CDM	*A-LINK*		
nS6	6	SetAttribute	Data Exchange
nS7	7	Restore Default	Directive

#1 Prefix n in CDM service ID represents one of Sai, Sei, Sbi, Sbith, Aao, Aeo, Abo and C.

8 Protocol Compliance

8.1 A method to testing protocol compliance is required to verify implementation conformance to this standard. A-LINK User Forum (AUF) has established a qualified certification mechanism of conformance testing and interoperability testing. The first laboratory is constituted in Japan. A-LINK conformance test information can be found at http://www.a-linkuf.com.

9 Specific Device Model Mappings

9.1 Every type of device must have an identifier number. Vendors must apply for an identifier number from the A-LINK User Organization for every Device Type.

9.1.1 The Device Profile must specify the identifiers for Objects, Attributes and Services for CDM and SDM components, including data formats and bit mappings for specified parameters, as represented in this document.

9.1.2 The following sections specify mappings for Sensor Actuator Network Specific Device Models.

9.2 *Mass Flow Device*

9.2.1 *MFD Device* — Reference SEMI E54.3 for a complete specification of the SDM for Mass Flow Devices. Accordingly, the following mapping rules apply to the identification tags for the Objects, Attributes and Services of this model.

9.2.2 *Objects* — Consistent with SEMI E54.3 and ¶7.4 above, the DM and SAC objects are identified as Object

9.2.2.1 *Mapping* — Table 13 shows the mapping of the SDM Object specified in SEMI E54.3.

Table 13 MFD Object Identifiers

SDM Object Name	*SDM Object ID*	*ID*	*Instance*
Device Manager (DM)	MFD1	1	1
Sensor Actuator Contriller (SAC)	MFD2	2	1
Sensor-AI-MF	MFD3	34	1 or supplier specific
Sensor-AI-AT	MFD4	35	0 / 0 or supplier specific
Assembly-MFM	MFD5	32	0 or 1
Sensor-AI-Aux	MFD6	36	0 / 0 or supplier specific
Actuator-AO-MF	MFD7	37	0 / 1 or supplier specific
Controller	MFD8	24	0 / 1
Local Link	MFD9	4	0 / 2 or supplier specific
SISO	MFD10	38	0 / 0 or supplier specific
SISO-Setpoint	MFD11	39	0 / 0 or supplier specific
Assembly-MFC	MFD12	33	0 / 0 or supplier specific

#1 / = numbers for MFM / MFC

9.2.3 *Attributes and Services*

9.2.3.1 *Attributes* — The mapping of Attribute tags and Identifiers is defined in ¶7.5.3.2 for the CDM. Basically the same definitions are applied for attributes in the Mass Flow Device SDM.

9.2.3.2 *Services* — The mapping of Service tags and Identifiers is defined in ¶7.6.2.2 for the CDM. Basically the same definitions are applied for attributes in the Mass Flow Device SDM.

9.2.3.3 *MFD Specific Attributes and Services* — Some objects appear in MFD SDM use additional attributes and/or additional services.

9.2.3.3.1 *Sensor AI-MF Object* — Additional attributes and services are mapped as following tables.

Table 14 Sensor AI-MF Object Attribute Identifiers

Attribute ID		*Attribute Name*
SDM	*A-LINK*	
A1	192	Flow Totalizer
A2	193	Flow Hours
A5	196	Zero Offset Mode
A6	197	Zeroing Status
A7	198	Autorange Status

Table 15 Sensor AI-MF Object Service Identifiers

Service ID		*Service Name*
SDM	*A-LINK*	
S1	16	Perform Zero Offset
S2	17	Query Supported Gas type
S3	18	Select Programmed Gas Type
S4	19	Insert Gas Type
S5	20	Delete Gas Type
S6	21	Get Gas Calibration Data Value
S7	22	Set Gas Calibration Data Value
S8	23	Autorange

9.2.3.3.2 *Actuator AO-MF Object* — Additional attributes are mapped as following table.

Table 16 Actuator AO-MF Object Attribute Identifiers

Attribute ID		*Attribute Name*
SDM	*A-LINK*	
A1	192	Valve Type
A2	193	Override

9.2.3.3.3 *Controller Object* — This object has no additional attribute and/or service but form of following a couple of attributes is restricted as Real.

9.2.3.3.4 *SISO Object* — This object is specific for MFD SDM. Attributes of the object are mapped as following table.

Table 17 SISO Object Attribute Identifiers

Attribute ID		*Attribute Name*
SDM	*A-LINK*	
A1	192	Input
A2	193	Output
A3	194	Data Type

9.2.3.3.5 *SISO Setpoint Object* — This object is specific for MFD SDM and inherits SISO object. Attributes of the derived object are mapped as following table.

Table 18 SISO Setpoint Object Attribute Identifiers

Attribute ID		*Attribute Name*
SDM	*A-LINK*	
A33	224	Ramp type
A34	225	Ramp Rate
A35	226	Ratio

9.3 *In-Situ Particle Monitor*

9.3.1 *ISPM Device* — Reference SEMI E54.10 for a complete specification of the SDM for In-Situ Particle Monitor Devices. Accordingly, the following mapping rules apply to the identification tags for the Objects, Attributes and Services of this model.

9.3.2 *Objects* — Consistent with SEMI E54.10 and ¶7.4 above, the DM and SAC objects are identified as Object 1 and Object 2, respectively.

9.3.2.1 *Mapping* — Table 19 shows the mapping of the SDM Object Instances specified in SEMI E54.10 (Instance numbers are listed under heading Inst. in the table) and the A-LINK Object ID (listed under ID in the table).

Table 19 ISPM Object Identifiers

SDM Object Name	*SDM Object ID*	*ID*	*Instance*
Device Manager (DM)	ISPMD1	1	1
Sensor Actuator Controller (SAC)	ISPMD2	2	1
Sensor-AI-LCS	ISPMD3	32	0 or 1
Sensor-AI-SLS	ISPMD4	33	0 or 1
Sensor-AI-MNS	ISPMD5	34	0 or 1
Sensor-AI-Counter	ISPMD16	45	0 or n*
Assembly-ISPM#1	ISPMD17	35	1
Assembly-ISPM#2	ISPMD18	36	0 or 1
Assembly-ISPM#3	ISPMD19	37	0 or 1
Assembly-ISPM#4	ISPMD20	38	0 or 1
Assembly-ISPM#5	ISPMD21	39	0 or 1
Assembly-ISPM#6	ISPMD22	40	1
Assembly-ISPM#7	ISPMD23	41	0 or 1
Assembly-ISPM#8	ISPMD24	42	0 or 1
Assembly-ISPM#9	ISPMD25	43	0 or 1
Assembly-ISPM#48	ISPMD64	44	0 or 1

#1 Minimum number of the n is one and possible maximum number is 1024, dependent on supplier.

9.3.2.2 Additional objects may be defined by the manufacturer in the Device Profile for a given device.

9.3.3 *Attributes and Services*

9.3.3.1 *Attributes* — The mapping of Attribute tags and Identifiers is defined in ¶7.5.3.2 for the CDM. Basically the same definitions are applied for attributes in the In-Situ Particle Monitor SDM.

9.3.3.2 *Services* — The mapping of Service tags and Identifiers is defined in ¶7.6.2.2 for the CDM. Basically the same definitions are applied for attributes in the In-Situ Particle Monitor SDM.

9.3.3.3 *ISPM Specific Attributes and Services* — Some objects appear in ISPM SDM use additional attributes and/or additional services.

9.3.3.3.1 *DM Object* — Additional attributes and services are mapped as following tables.

Table 20 DM Object Attribute Identifiers

Attribute ID		*Attribute Name*
SDM	*A-LINK*	
A33	33	Gain
A34	34	Filter Bandwidth
A35	35	Tool State
A36	36	Laser Status
A37	37	Flow Path
A38	38	Volume
A39	39	Volume Units
A40	40	Leak Status
A41	41	Time Stamp

Table 21 DM Object Service Identifiers

Service ID		*Service Name*
SDM	*A-LINK*	
S1	33	Laser On
S2	34	Laser Off

9.3.3.3.2 *SAC Object* — Additional attributes and services are mapped as following tables.

Table 22 SAC Object Attribute Identifiers

Attribute ID		*Attribute Name*
SDM	*A-LINK*	
SacA65	33	Number of Bins
SacA66	34	Count Mode
SacA67	35	Duration

Table 23 SAC Object Service Identifiers

Service ID		*Service Name*
SDM	*A-LINK*	
S33	33	Clear Counts

9.3.3.3.3 *Sensor AI-LCS Object* — Additional attributes are mapped as following table.

Table 24 Sensor AI-LCS Object Attribute Identifiers

Attribute ID		*Attribute Name*
SDM	*A-LINK*	
LcsA1	192	Reading Valid
LcsA2	193	Full Scale
LcsA3	194	Alarm Setting Time
LcsA4	195	Warning Setting Time

9.3.3.3.4 *Sensor AI-SLS Object* — Additional attributes are mapped as following table.

Table 25 Sensor AI-SLS Object Attribute Identifiers

Attribute ID		*Attribute Name*
SDM	*A-LINK*	
SlsA1	192	Reading Valid
SlsA2	193	Full Scale
SlsA3	194	Alarm Setting Time
SlsA4	195	Warning Setting Time

9.3.3.3.5 *Sensor AI-MNS Object* — Additional attributes are mapped as following table.

Table 26 Sensor AI-MNS Object Attribute Identifiers

Attribute ID		*Attribute Name*
SDM	*A-LINK*	
MnsA1	192	Reading Valid
MnsA2	193	Full Scale
MnsA3	194	Alarm Setting Time
MnsA4	195	Warning Setting Time

9.3.3.3.6 *Sensor AI-Counter Object* — Additional attributes are mapped as following table.

Table 27 Sensor AI-Counter Object Attribute Identifiers

Attribute ID		*Attribute Name*
SDM	*A-LINK*	
CounterA1	192	Reading Valid
CounterA2	193	Full Scale
CounterA3	194	Alarm Setting Time
CounterA4	195	Warning Setting Time
CounterA5	196	Upper Size
CounterA6	197	Lower Size

9.4 *Endpoint Detector*

9.4.1 *EPD Device* — Reference SEMI E54.11 for a complete specification of the SDM for Endpoint Devices. Accordingly, the following mapping rules apply to the identification tags for the Objects, Attributes and Services of this model.

9.4.2 *Objects* — Consistent with SEMI E54.11 and ¶7.4 above, the DM and SAC objects are identified as Object 1 and Object 2, respectively.

9.4.2.1 Table 28 shows the mapping of the SDM Object Instances specified in SEMI E54.11 (Instance numbers are listed under heading Inst. in the table) and the A-LINK Object ID (listed under ID in the table).

Table 28 EPD Object Identifiers

SDM Object Name	*SDM Object ID*	*ID*	*Instance*
Device Manager (DM)	EPD1	1	1
Sensor Actuator Controller (SAC)	EPD2	2	1
Sensor-BI-TH-EP	EPD3	36	0 or n*
Assembly-EPD#1	EPD4	32	1
Assembly-EPD#2	EPD5	33	0 or 1
Assembly-EPD#3	EPD6	34	0 or 1
Assembly-EPD#4	EPD7	35	0 or 1

#1 Minimum number of the n is one and possible maximum number is 1024.

9.4.2.2 Additional objects may be defined by the manufacturer in the Device Profile for a given device.

9.4.3 *Attributes and Services*

9.4.3.1 *Attributes* — The mapping of Attribute tags and Identifiers is defined in ¶7.5.3.2 for the CDM. Basically the same definitions are applied for attributes in the Endpoint Detector SDM.

9.4.3.2 *Services* — The mapping of Service tags and Identifiers is defined in ¶7.6.2.2 for the CDM. Basically the same definitions are applied for attributes in the Endpoint Detector SDM.

9.4.3.3 *EPD Specific Attributes and Services* — Some objects appear in EPD SDM use additional attributes and/or additional services.

9.4.3.3.1 *SAC Object* — Additional attributes and services are mapped as following tables.

Table 29 SAC Object Attribute Identifiers

Attribute ID		*Attribute Name*
SDM	*A-LINK*	
SacA65	33	Number of Endpoint Objects

Table 30 SAC Object Service Identifiers

Service ID		*Service Name*
SDM	*A-LINK*	
S33	33	Reset Endpoint
S34	34	Download Recipe
S35	35	Upload Recipe
S36	36	Calibrate

9.4.3.3.2 *Sensor-BI-TH-EP Object* — Additional attributes and services are mapped as following tables.

Table 31 Sensor-BI-TH-EP Object Attribute Identifiers

Attribute ID		*Attribute Name*
SDM	*A-LINK*	
EpA1	192	Minimum Time
EpA2	193	Maximum Time
EpA3	194	Target Time
EpA4	195	Elapsed Time
EpA5	196	Time Stamp
EpA6	197	Recipe Identifier
EpA7	198	Step Identifier

Table 32 Sensor-BI-TH-EP Object Service Identifiers

Service ID		*Service Name*
SDM	*A-LINK*	
S1	16	Endpoint On
S2	17	Endpoint Off
S3	18	Endpoint Start
S4	19	Endpoint Suspend
S5	20	Endpoint Resume

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 E58-0703
AUTOMATED RELIABILITY, AVAILABILITY, AND MAINTAINABILITY STANDARD (ARAMS): CONCEPTS, BEHAVIOR, AND SERVICES

This standard 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 October 19, 2000. Initially available at www.semi.org January 2001; to be published March 2001. Originally published June 1997.

NOTICE: The designation of SEMI E58 was updated during the 0703 publishing cycle to reflect the reapproval of SEMI E58.1.

CONTENTS

1 Purpose

1.2 Background and Motivations

2 Scope

3 Referenced Documents

4 Terminology

4.1 Acronyms

4.2 General Terms

4.3 Data Types

5 Basic Requirements

6 Conventions

6.1 State Model Methodology

6.2 Object Attribute Representation

6.3 Service Message Representation

6.3.1 Service Resource Definition

6.3.2 Service Parameter Dictionary

6.3.3 Service Message Definition

7 Overview

8 State Models

8.2 SEMI E10 Equipment States

8.3 ARAMS State Model Definition

8.3.1 ARAMS State Model Diagram

8.3.2 Descriptions of ARAMS States

8.3.2.1 TOTAL TIME

8.3.2.2 MANUFACTURING

8.3.2.3 PRODUCTIVE

8.3.2.4 STANDBY

8.3.2.5 ENGINEERING

8.3.2.6 UNSCHEDULED DOWNTIME

8.3.2.7 SCHEDULED DOWNTIME

8.3.2.8 NON-SCHEDULDED TIME

8.3.3 ARAMS Substates

8.3.4 State Transitions

8.4 General Equipment Operations Model

8.4.2 State Definitions

8.4.2.1 POWER-OFF

8.4.2.2 POWER-ON

8.4.2.3 INITIALIZING

8.4.2.4 READY

8.4.2.5 IDLE

8.4.2.6 BUSY

8.4.3 General Equipment Operations Model Table of Transitions

8.5 State Model Relationships

8.5.2 IDLE and BUSY

8.5.3 Non-Manufacturing States

9 ARAMS Substate Codes

9.2 Reserved Codes

9.3 Additional Codes

9.4 Valid ARAMS Substate Code

9.5 Manufacturing Code

10 ARAMS Tables

10.2 Definition of Tables

10.2.1 Table Types and Identifiers

10.2.2 Table Row Definition

10.2.3 Table Attributes

10.2.4 Additional Requirements

10.3 ARAMS Substate Table

10.4 ARAMS Symptom Table
11 ARAMS Data
11.2 Status Information
11.2.3 ARAMSInfo
11.2.4 ARAMSState
11.2.5 ARAMSText
11.2.6 Clock
11.2.7 CycleCtr
11.2.8 DowntimeAlarm
11.2.9 DowntimeAlarmText
11.2.10 DowntimeData
11.2.11 LastPowerdown
11.2.12 PowerdownTime
11.2.13 PrdState
11.2.14 PrevARAMSState
11.2.15 SymptomID
11.2.16 SymptomText
11.3 Constant Data
11.3.1 EqpModel
11.3.2 EqpSerialNum
11.4 User-Configurable Data
11.4.1 EngInterrupt (optional)
11.4.2 EngRecovery (optional)
11.4.3 EqpName (required)
11.4.4 PowerupState (optional)
11.4.5 PrdRecovery (optional)
11.4.6 SbyRecovery (optional)
11.4.7 SubstateSelect (optional)
11.5 Accumulators
11.5.2 ARAMSAccumReset
11.5.3 ARAMSTimestamp
11.5.4 EngTime
11.5.5 InterruptionPrd
11.5.6 InterruptionTotal
11.5.7 NSTime
11.5.8 PrdTime
11.5.9 SbyTime
11.5.10 SDTime
11.5.11 UDTime
12 Events and Pre-Defined Event Reports
13 Object Services Compliance
13.1 Equipment Object
13.2 Table Objects
14 Human Interface Requirements
14.1 Data Access
14.2 Selection of an ARAMS State/Substate
14.3 User-Defined ARAMS Symptom Tables
14.4 Table Access
14.5 Color Codes
15 Services
15.2 Services Parameter Dictionary
15.3 TableSend
15.4 TableRequest
15.5 ARAMSStateChange
15.6 ResetAccumulators
16 ARAMS Behavioral Requirements
16.1 ARAMS State Transitions
16.2 Powerup Entry
16.3 Powerup and Powerdown States
16.3.2 PowerupState
16.4 User-Initiated State Change Requests
16.5 Production and Standby Substates
16.6 Equipment-Selected Substates
16.7 Equipment-Detected Exceptions
16.7.4 Fault Detection in ENGINEERING
16.8 Equipment-Initiated Recovery
16.8.2 Automatic Recovery
17 Requirements for Compliance
17.1 Fundamental Requirements
17.1.1 Event Notification
17.1.2 Clock Services
17.1.3 Read-Only Data Access
17.1.4 User-Configurable Data Access
17.1.5 Alarm/Exception Management

17.1.6 ARAMS State Model

17.1.7 ARAMS State Transition Notification

17.1.8 ARAMS Substate Codes

17.1.9 ARAMS Status Data

17.1.10 ARAMS Constant Data

17.1.11 ARAMS Event Report Data

17.1.12 Host State Change Request

17.1.13 Estimation of Powerdown Time

17.1.14 ARAMS Behavioral Requirements

17.2 Additional Capabilities

17.2.1 User-Configurable Powerup State

17.2.2 User-Configurable Fault Recovery to Manufacturing

17.2.3 Accumulator Data

17.2.4 User-Generated ARAMS Substate Table(s)

17.2.5 Equipment-Generated ARAMS Substate Table(s)

17.2.6 User-Generated Symptom Table(s)

17.2.7 Human Interface Requirements

17.2.8 Equipment-Selected Substates

17.2.9 User-Configurable Fault Detection in ENGINEERING

17.2.10 User-Configurable Fault Recovery to ENGINEERING

17.2.11 Human Interface Requirements

17.3 Requirements for Compliance

18 ARAMS States for Multi-Module Equipment

Related Information 1

R1-1 Estimating Powerdown Time

R1-1.2 UpdatePeriod

R1-2 Powerup Scenario

R1-3 Equipment-Initiated Transition

R1-4 Operator-Initiated Transition

R1-5 Host-Initiated Transition

Related Information 2

R2-1 User-Initiated Transitions to UNSCHEDULED DOWNTIME

SEMI E58-0703
AUTOMATED RELIABILITY, AVAILABILITY, AND MAINTAINABILITY STANDARD (ARAMS): CONCEPTS, BEHAVIOR, AND SERVICES

This standard 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 October 19, 2000. Initially available at www.semi.org January 2001; to be published March 2001. Originally published June 1997.

NOTICE: The designation of SEMI E58 was updated during the 0703 publishing cycle to reflect the reapproval of SEMI E58.1.

1 Purpose

1.1 This document provides standards for implementing and collecting SEMI E10 state changes at the equipment level per SEMI E10.

1.1.1 SEMI E10 defines various terms and equipment states but was not written specifically for application by automated equipment. This document is intended to provide a consistent interpretation of these equipment states through formal state model methodology.

1.1.2 ARAMS defines concepts, behavior, and message services to support the integration of automated systems within a semiconductor factory.

1.1 *Background and Motivations* — To implement the integration of SEMI E10 states on automated equipment, integration of definitions and requirements must be detailed and precise to ensure interpretations are consistent across equipment suppliers. This provides an opportunity to automatically retain information at the equipment itself.

1.1.3 Both equipment supplier and equipment user benefit from the automation of SEMI E10 data collection at the equipment through application of a consistent state model.

1.1.4 SEMI E10 defines specific states but does not address transitions between states. The ARAMS standard specifies the triggers for state transitions made by automated equipment. Extensions to SEMI E10 described in this document apply to decisions made by automated equipment only.

2 Scope

1.2 This standard is applicable to the following relationships: traditional host/equipment, operator/equipment, and cluster tool controller/attached module. The scope of this document is to define standards which facilitate equipment-level capture and communication of SEMI E10 related data. Specifically, this document provides the following:

- An equipment state model that defines the rules for equipment state changes,
- A set of standard equipment codes for representing substates of the six basic equipment states defined in SEMI E10,
- Definition of equipment-generated data,
- Concepts and messages required to exchange information, and
- Requirements for fundamental compliance to ARAMS
- Additional optional specifications.

1.3 This standard is intended as a supplement to SEMI E10 to be used for equipment support of SEMI E10. Formal definitions of all terms common to both documents are provided solely by SEMI E10.

1.4 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

3.1 *SEMI Standards*

SEMI E10 — Standard for Definition and Measurement of Equipment Reliability, Availability, and Maintainability (RAM)

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

SEMI E38 — Cluster Tool Module Communications (CTMC)

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

SEMI E41 — Exception Management (EM) Standard

SEMI E42 — Recipe Management Standard: Concepts, Behavior, and Message Services

SEMI E53 — Event Reporting

3.2 *Other Document*

Harel, D., "Statecharts: A Visual Formalism for Complex Systems," *Science of Computer Programming 8* (1987) 231–274

NOTE 1: As listed or revised, all documents cited shall be the latest publications of adopted standards.

4 Terminology

4.1 *Acronyms* — The following acronyms are used in this document.

4.1.1 *ARAMS* — Automated Reliability, Availability, and Maintainability Standard, as defined by this document.

4.1.2 *CTMC* — Cluster Tool Module Communications [SEMI E38].

4.1.3 *EMS* — Exception Management Standard [SEMI E41].

4.1.4 *ERS* — Event Reporting Standard [SEMI E53].

4.1.5 *GEM* — Generic Equipment Model [SEMI E30].

4.1.6 *OSS* — Object Services Standard [SEMI E39].

4.1.7 *RAM* — Reliability, Availability, and Maintainability.

4.2 *General Terms* — The following definitions for general terms are used in this document. References are given in brackets.

4.2.1 *alarm* — Related to any abnormal situation on the equipment that may endanger people, equipment, or material being processed [SEMI E30, SEMI E41].

4.2.2 *collection event* — An event (or grouping of related events) on the equipment that is considered to be significant to the host [SEMI E30].

NOTE 2: A state transition in a formal state model always represents a collection event unless explicitly stated otherwise.

4.2.3 *equipment production criteria* — The set of conditions and operating specifications that must be satisfied for the equipment to consider itself as performing its intended function. This includes basic requirements for information, material to process, and the absence of any detectable exception conditions (e.g., no alarms). It also includes criteria specific to the equipment model, such as a required level for vacuum pressure and availability of consumables and support tools required for its process.

4.2.4 *event* — A detectable occurrence significant to the equipment.

NOTE 3: Within the context of ARAMS, an event may be detected by either the equipment or the user.

4.2.5 *event report* — A message the equipment sends to the host on the occurrence of a collection event.

4.2.6 *exception* — An alarm or error that is reported to the user and that may or may not be recoverable.

4.2.7 *fault* — An exception.

4.2.8 *host* — The intelligent system that communicates with the equipment, acts as a supervisory agent, and represents the factory and the user to the equipment.

4.2.9 *intended function* — A manufacturing function that the equipment was built to perform. This includes transport functions for transport equipment and measurement functions for metrology equipment as well as process functions such as physical vapor deposition and wire bonding. Complex equipment may have more than one intended function.

4.2.10 *interrupt (interruption)* — A failure [SEMI E10].

4.2.11 *operator* — Any person who communicates locally with the equipment through the equipment's control panel.

4.2.12 *state* — A static set of conditions and associated behavior. While all of its conditions are met, the state is current (active). Behavior within a given state includes the response to various stimuli.

NOTE 4: Within the scope of this document, the term "state" generally refers to one of the six equipment states defined by SEMI E10 and used in the ARAMS State Model: productive, standby, engineering, scheduled downtime, unscheduled downtime, and non-scheduled time.

4.2.13 *state model* — A collection of states and state transitions that combine to describe the behavior of a system. This model includes a definition of the conditions that delineate a state, the activities possible within a state, the events that trigger transitions to other states, and the process of transitioning between states.

4.2.14 *state transition* — A change from one state to another state.

4.2.15 *standby condition* — Any condition during manufacturing time when the equipment's production criteria are not satisfied, and it is fault free and otherwise able to perform its intended function.

4.2.16 *substate* — A refinement of a state.

NOTE 5: States may be subdivided into substates to facilitate more concise definition of behavior. Thus, a hierarchy is defined whereby any state may be a substate of some parent state and in turn be the parent of its own substates [SEMI E30, Appendix].

4.2.17 *superstate* — The parent state of two or more states.

4.2.18 *symptom* — A user-detected event (e.g., smoke observed).

4.2.19 *timestamp* — The notation of the date and time of the occurrence of an event [SEMI E42].

4.2.20 *timestamp format* — A text string of the form “YYYYMMDDhhmmsscc”, where:

YYYY = year (e.g., 1995)

MM = month (01–12)

DD = day (01–31)

hh = hour (00–23)

mm = minute (00–59)

ss = second (00–59)

cc = centisecond (00–99)

4.2.21 *trigger* — An event that causes a change in the state of the equipment. Examples are changes in sensor readings, alarms, messages received from the host, and operator commands.

4.2.22 *user* — Any entity interacting with the equipment, either locally as an operator or remotely via the host. From the equipment’s viewpoint, both the operator and the host represent the user.

4.3 *data types* — The following terms are used to represent valid types of data.

4.3.1 *form* — Type of data: positive integer, unsigned integer, integer, floating point (float) enumerated, Boolean, text, formatted text, structure, list, ordered list.

4.3.2 *positive integer* — May take the value of any positive whole number. Messaging protocol may impose a limit on the range of possible values.

4.3.3 *unsigned integer* — May take the value of any positive integer or zero. Messaging protocol may impose a limit on the range of possible values.

4.3.4 *integer* — May take on the value of any negative or unsigned integer. Messaging protocol may impose a limit on the range of possible values.

4.3.5 *floating point (float)* — May take on any single (real) numeric value, positive or negative. Messaging protocol may impose a limit on the range of possible values.

4.3.6 *enumerated* — May take on one of a limited set of possible values. These values may be given logical names, but they may be represented by any single-item data type.

4.3.7 *boolean* — May take on one of two possible values, equating to TRUE and FALSE.

4.3.8 *text* — A character string. Messaging protocol may impose restrictions, such as length or ASCII representation.

4.3.9 *formatted text* — A character string with an imposed format. This could be by position, by use of special characters, or both.

4.3.10 *structure* — A specific set of items, of possibly mixed data types, in a specified arrangement.

4.3.11 *list* — A set of one or more items that are all of the same form (one of the above forms).

4.3.12 *ordered list* — A set of items in specific sequence.

5 Basic Requirements

1.5 An ARAMS-compliant implementation requires provision of certain capabilities defined by other standards: accessibility to status information, event reporting, alarm management, and provision of an internal time-and-date clock. These requirements may be satisfied through compliance to one of the following sets of requirements:

- The Generic Equipment Model (GEM):
  - Clock Services
  - Event Notification
  - Status Data Collection
  - Equipment Constants
  - Alarm Management
- The following set of standards:
  - Object Services Standard
  - Clock Services, Cluster Tool Module Communications
  - Event Reporting Standard
  - Exception Management Standard

1.6 The developer is expected to be familiar with the appropriate documents before attempting to implement ARAMS (see Section 16.1).

6 Conventions

This document follows the conventions for state model methodology and service definitions used by the SEMI standards referenced in Section 3.

6.1 *State Model Methodology* — This document uses the state model methodology in SEMI E30 to describe the behavior of equipment. A state model has three elements: definitions of each state and substate, a diagram of the states and the transitions between states, and a state transition table. The diagram of the state model uses the Harel State Chart notation. An overview of this notation is presented in an appendix of SEMI E30. The formal definition of this notation is presented in *Science of Computer Programming 8*, "Statecharts: A Visual Formalism for Complex Systems", by D. Harel, 1987.

1.6.1 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 #, Current State, Trigger, New State, Action(s), and Comment. The "trigger" (column 3) for the transition occurs while in the "current" state. The "actions" (column 5) includes a combination of 1) actions taken upon exit of the current state, 2) actions taken upon entry of the new state, and 3) actions taken which are most closely associated with the transition. No differentiation is made between these cases.

#	*Current State*	*Trigger*	*New State*	*Action(s)*	*Comment*
Transition #					

1.7 *Object Attribute Representation* — The object information models for standardized objects will be supported by an **attribute definition table** with the following column headings:

Attribute Name	*Definition*	*Access*	*Reqd*	*Form*
The formal text name of the attribute.	Description of the information contained.	RO or RW	Y or N	(see below)

1.7.1 The Access column uses RO (Read Only) or RW (Read and Write) to indicate the access that users of the service have to the attribute.

1.7.2 A 'Y' or 'N' in the Required (Reqd) column indicates if this attribute must be supported in order to meet fundamental compliance for the service.

1.7.3 The Form column is used to indicate the format of the attribute (see Section 4 for definitions).

6.2 *Service Message Representation*

1.7.4 *Service Resource Definition* — 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*
Message name	N or R	The intent of the service

1.7.4.1 Type can be either N = Notification or R = Request.

1.7.4.2 Notification type messages are initiated by the service provider and the provider does not expect to get a response from the service user (consumer/subscriber).

1.7.4.3 Request messages are initiated by a service user. Request messages ask for data or an activity from the provider. Request messages expect a specific response message (no presumption on the message content).

1.7.5 *Service Parameter Dictionary* — A **service parameter dictionary table** defines the parameters for one or more services, as shown in the following table:

Parameter	*Description*	*Form*
Parameter X	A parameter called X is B in A.	Data type.

1.7.5.1 A row is provided in the table for each parameter of the service. The first column contains the name of the parameter. This is followed by columns describing the form and contents of the corresponding parameter.

1.7.5.2 The Form column is used to indicate the type of data contained in a parameter (see Section 4 for definitions).

1.7.5.3 The Description column in the Service Parameter Dictionary table describes the meaning of the parameter, the allowed values, and any interrelationships with other parameters.

1.7.5.4 To prevent the definition of numerous parameters named "XxxList", this document adopts the convention of referring to the list as "(List of) Xxx". In this case, the definition of the variable Xxx will be given, not of the list. The term "list" indicates a collection (or set) of zero or more items of the same data type.

1.7.5.5 Where a list is used in both the request and the response, the list order in the request is retained in the response. A list must contain at least one element unless zero elements are specifically allowed.

1.7.6 *Service Message Definition* — A **service message definition table** defines the parameters used in a service, as shown in the following table:

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
Parameter X	(see below)	(see below)	A description of the parameter.

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

1.7.6.2 The following codes appear in the Req/Ind and Rsp/Conf 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 a completely optional or may depend on the value of other parameters.

“U” - *User-defined parameter*

“-” — The parameter is not used.

“=” — (For response only) Indicates that the value of this parameter in the response must match that in the primary (if defined).

7 Overview

1.8 This section provides an overview of how ARAMS will be applied and the capabilities that ARAMS defines.

1.9 Systems that are used to track equipment performance should be based on SEMI E10’s definitions of the six basic equipment states. The tracking systems typically rely on factory personnel to manually enter SEMI E10 state changes. Individual factories may have further company-specific and/or facilities-specific refinements of the states. For example, the basic equipment state may be “unscheduled downtime” with a refinement of “waiting for parts”. In addition to entering a state change, the operator (who may be the production operator, a process engineer, an equipment engineer, or a supplier field service engineer) may select from a pre-defined set of behaviors (symptoms) that prompted the change of state, such as “smoke observed”.

1.10 Specifications provided by ARAMS are intended to support the integration of equipment systems with factory tracking systems. For this purpose, equipment needs to be cognizant of the ARAMS states and substates, must know its current state, and must follow common rules for determining if equipment-initiated transitions can be made. In addition, the user needs to be able to interact with the tracking system at either the host system’s console or the equipment’s operator console. This requires that the operator be able to enter certain information at the equipment’s console: a new state or substate request, and specific observed behavior that prompted the request. The host system is then notified that a change in state has occurred.

1.11 Integrated systems are able to provide more accurate data for those state changes than the equipment alone is able to detect. While the user is still required to initiate state changes for other conditions, this can be accomplished either directly at the equipment’s console or remotely at the host tracking system terminal.

1.12 Exchange of information is accomplished through standardization of the meaning and form of data and the specification of the message services for the exchange. ARAMS provides generic definitions for the common substates described in SEMI E10. ARAMS also defines two tables and the message services required for the equipment and host to exchange tables. The first table contains a set of ARAMS substate definitions (an ARAMS code identifying the substate) and a corresponding description. The second table defines a set of symptoms with a numeric symptom identifier and a corresponding description.

1.13 While the above discussion assumes that the equipment is interacting with host systems, the ARAMS state model only requires interactions with a “user”, which might be either a local operator or a host system.

8 State Models

1.14 This section defines the formal state model for ARAMS, called the ARAMS State Model, which is required for ARAMS compliance. To clarify the relationships between this state model and the equipment’s operations, it introduces a second state model, called the General Equipment Operations Model, which is used for purposes of illustration. The General Equipment Operations Model is assumed to exist in some form but is not required for ARAMS compliance.

1.14.1 This document follows the convention of using upper-case to denote the formal names of states. Informal references may use lower case. For example, the ARAMS states SCHEDULED DOWNTIME and UNSCHEDULED DOWNTIME may be referred to as downtime states.

1.14.2 Detailed requirements for equipment behavior are provided in Section 16.

NOTE 6: Although the equipment is unable to detect a condition of no power, it is able to detect when the INITIALIZING state has been entered and is able to differentiate between a hard reset (Transition 6) and a soft re-boot (Transition 5).

8.1 *SEMI E10 Equipment States* — Figure 1 contains a diagram of SEMI E10 equipment states using the Harel notation. SEMI E10 divides total time into six basic states: PRODUCTIVE, ENGINEERING, STANDBY, SCHEDULED DOWNTIME, UNSCHEDULED DOWNTIME, and NON-SCHEDULED TIME. These six states are shown in Figure 1 with solid lines.

1.14.3 OPERATIONS TIME, UPTIME, DOWNTIME, and MANUFACTURING TIME are derived by grouping states defined in SEMI E10 and are useful for classification purposes, but formally they are not considered as SEMI E10 equipment states. Time in these groupings can be derived by summing the time in their corresponding states, based on Figure 1.

NOTE 7: Figure 1 uses shadings to show derived states. It is not intended as a formal state model.

1.14.4 MANUFACTURING TIME includes time spent in PRODUCTIVE and STANDBY. UPTIME includes the time spent in MANUFACTURING TIME and ENGINEERING. DOWNTIME includes the time spent in SCHEDULED DOWNTIME and UNSCHEDULED DOWNTIME.

1.14.5 In SEMI E10, precise rules governing state transitions are not required. The ARAMS model, in contrast, is intended to be used by automated equipment capable of detecting internal conditions. Conditions for each valid state transition are defined, both those initiated by equipment and those determined by interactions between the user and the equipment.

8.2 *ARAMS State Model Definition* — This section contains the formal definition of the ARAMS State Model, consisting of three parts:

- A diagram of the ARAMS State Model (Figure 2), using Harel notation,
- a description of each state and the behavior of the equipment within that state, and
- a table of transitions (Table 1) showing the previous state before the transition, the trigger for the transition, the new state after the transition, a description of any actions to be taken upon entry, and comments concerning the new state.

8.2.1 *ARAMS State Model Diagram* — Figure 2 contains the diagram of the ARAMS State Model.

8.2.2 *Descriptions of ARAMS States* — This section provides brief descriptions of the basic states for model completeness.

NOTE 8: These are informal descriptions included for the completeness of the ARAMS State Model. They do not replace the formal definitions in SEMI E10.

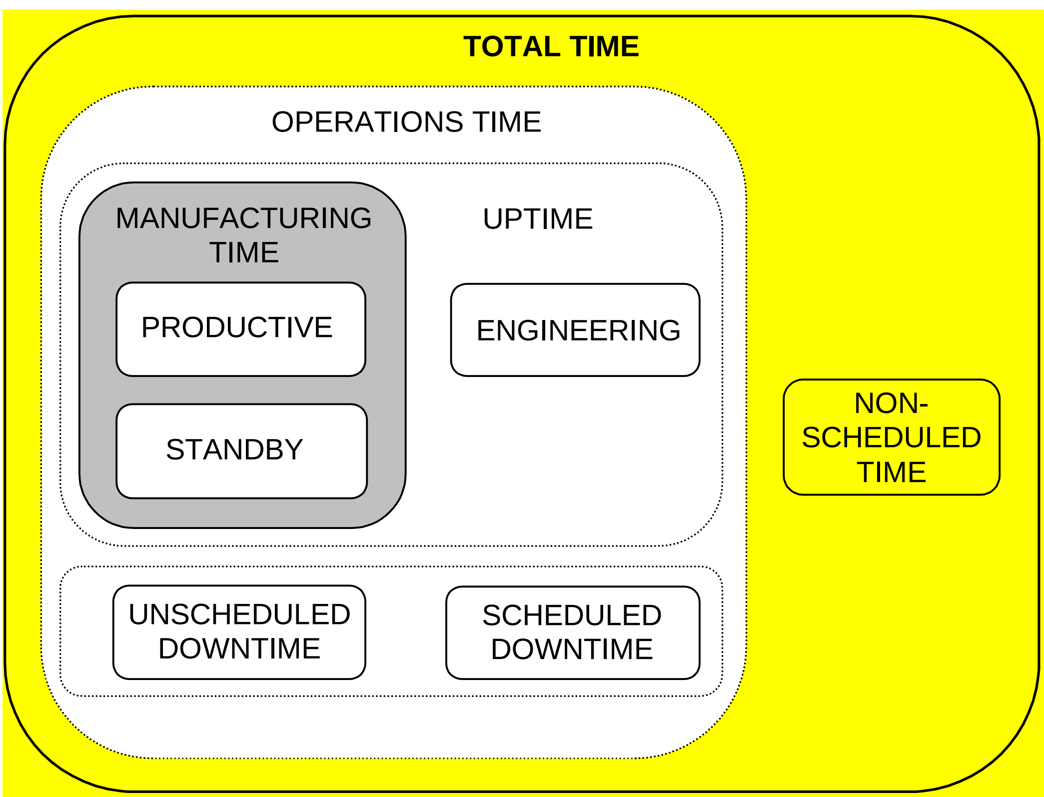

Figure 1

SEMI E10 Equipment States in Harel Notation

8.2.2.1 *TOTAL TIME* — The TOTAL TIME state includes 100% of real time; the sum of the time in the six basic SEMI E10 states, including time when the equipment is powered down. PRODUCTIVE, STANDBY, and ENGINEERING are called *uptime*, and SCHEDULED DOWNTIME and UNSCHEDULED DOWNTIME are called *downtime.*

8.2.2.2 *MANUFACTURING* — The ARAMS State Model includes MANUFACTURING as a user-selectable superstate of PRODUCTIVE and STAND-BY. When the user selects MANUFACTURING, the equipment automatically transitions to either PRODUCTIVE or STANDBY, depending upon its internal status at the time.

1.14.5.1.1 Equipment is fault-free during MANUFACTURING.

NOTE 9: MANUFACTURING is not a SEMI E10 equipment state.

8.2.2.3 *PRODUCTIVE* — The PRODUCTIVE state covers the time spent by the equipment in performing its intended function. This also includes time spent loading and unloading product. PRODUCTIVE is an uptime manufacturing state.

1.14.5.1.2 The equipment is in PRODUCTIVE when, and only when, it is in MANUFACTURING, its equipment production criteria are satisfied, and it is busy performing its intended function.

NOTE 10: Although by definition, the equipment is only considered to be "performing its intended function" in the PRODUCTIVE state, equipment processing cycles may occur in any of the basic SEMI E10 states except STANDBY.

8.2.2.4 *STANDBY* — The STANDBY state is an uptime manufacturing state that covers the time the equipment is waiting to enter the PRODUCTIVE state.

1.14.5.1.3 The equipment enters this state automatically from the PRODUCTIVE state whenever it is in the MANUFACTURING superstate and the requirements for PRODUCTIVE do not apply. This includes periods during which it detects a normal standby condition, such as no work, no operator, etc. During STANDBY, the equipment monitors conditions for PRODUCTIVE. When all requirements for PRODUCTIVE are satisfied, then it transitions automatically to PRODUCTIVE.

8.2.2.5 *ENGINEERING* — The ENGINEERING state is an uptime state that is selected by the user for process and equipment engineering purposes, such as process development or characterization.

1.14.5.1.4 Because the equipment may be pushed deliberately outside of its normal operating conditions, faults that may occur in the ENGINEERING state do not trigger equipment-initiated transitions to UNSCHEDULED DOWNTIME. The equipment may also be powered off while in ENGINEERING.

8.2.2.6 *UNSCHEDULED DOWNTIME* — The UNSCHEDULED DOWNTIME state is used for unplanned downtime activities, such as maintenance, setups, conversions, change of consumables, factory-related problems, etc.

1.14.5.1.5 Any transition from PRODUCTIVE to UNSCHEDULED DOWNTIME, whether equipment or user initiated, counts as a SEMI E10 failure. In some cases, where the equipment has detected an alarm condition and has transitioned to UNSCHEDULED DOWNTIME, the equipment is able to recover and return to PRODUCTIVE.

8.2.2.7 *SCHEDULED DOWNTIME* — The SCHEDULED DOWNTIME state is used for planned downtime activities, such as preventive maintenance, setups, conversions, change of consumables, factory-related events, etc.

8.2.2.8 *NON-SCHEDULED TIME* — The NON-SCHEDULED TIME state is used to account for time outside of the normal factory production schedule. This includes time when the factory itself is not operating and time when the equipment is being used for purposes other than production, engineering, or maintenance. Examples of such time include unworked shifts, holidays, plant shutdowns, installation, and off-line (outside of normal factory operations) training of personnel.

8.2.3 *ARAMS Substates* — Each of the six basic ARAMS states have refinements defined in SEMI E10. These refinements are captured by the ARAMS Substate Codes in Section 9. The host requests an ARAMS state change by specifying an ARAMS Substate Code directly, while the operator selects a state and substate combination, through the human interface, that results in an ARAMS Substate Code. The equipment then determines the appropriate ARAMS state/substate based on this code.

8.2.4 *State Transitions* — The user may ask the equipment to go to any ARAMS state at any time by specifying a new ARAMS Substate Code (see Section 9) or by specifying a code of "0000" to request a change to the MANUFACTURING superstate.

NOTE 11: A user-initiated ARAMS state change is not intended to initiate a change in the equipment's operation. For example, if the operator puts the equipment in UNSCHEDULED DOWNTIME while the equipment is completing a process cycle of material, the equipment shall complete its normal cycle. If the operator intends to abort the process, then the process must be specifically aborted.

1.14.5.2 Table 1 defines the triggers for each transition shown in Figure 2.

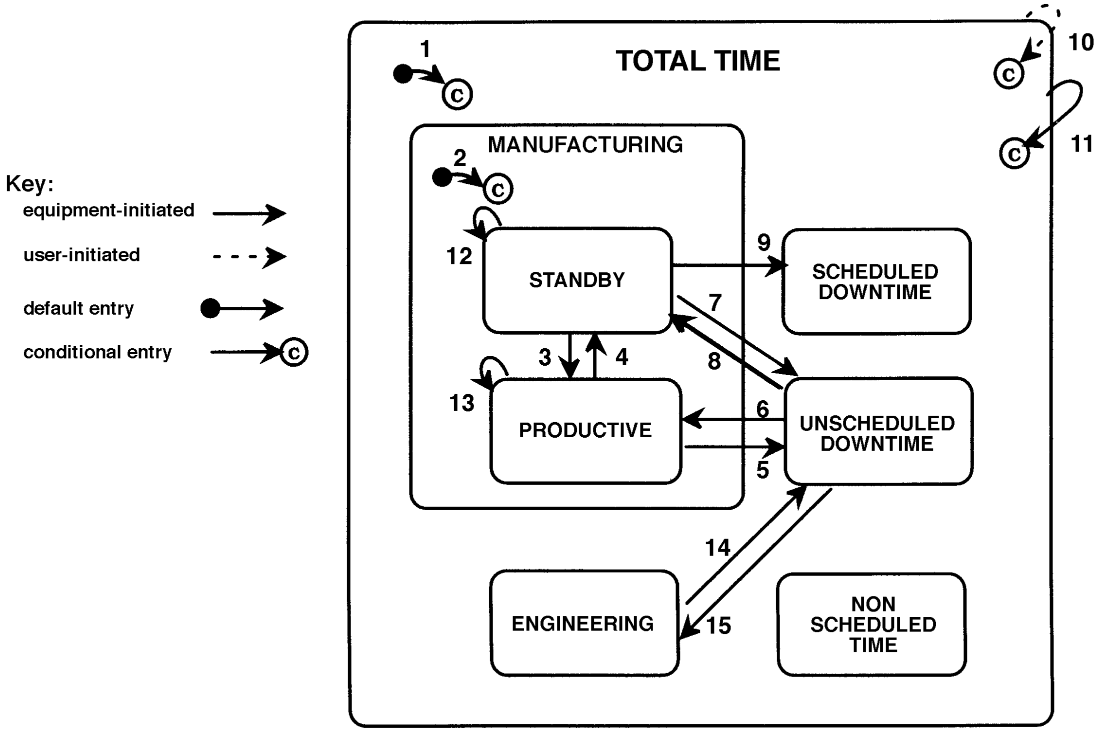

Figure 2
ARAMS State Model

Table 1 Table 1 Transitions for ARAMS State Model

#	*Current State*	*Trigger*	*New State*	*Action(s)*	*Comment*
1	(Either undeterminable or any state except PRODUCTIVE or STANDBY)	Powerup/reset	Depends upon the state in which Transition 11 occurred (see Section 8.5.2).	None	Entry state is dependent upon previous state (see NOTE 1) where this can be determined. May not generate an event report.
2	(any state)	User selects a manufacturing state.	PRODUCTIVE or STANDBY, depending upon the status of the equipment.	Determine the new ARAMS state/substate.	Equipment determines if production criteria are satisfied. If not, it transitions to STANDBY. No event report is generated.
3	STANDBY	Equipment detects that all its production criteria are satisfied.	PRODUCTIVE	Set ARAMSState to value in PrdState (Section 11.2).	May begin or resume processing.
4	PRODUCTIVE	Equipment detects a standby condition.	STANDBY	Monitor all production criteria.	Equipment may detect a standby condition at any time.
5	PRODUCTIVE	Equipment detects an exception.	UNSCHEDULED DOWNTIME	Increment InterruptionPrd (Section 11.5).	Alarm or exception report generated by same trigger.

#	*Current State*	*Trigger*	*New State*	*Action(s)*	*Comment*
6	UNSCHEDULED DOWNTIME	All fault conditions have been cleared.	PRODUCTIVE	Resume processing	Transition 6 can only follow Transition 5. Automatic recovery without operator direction may be disabled by the user. (See NOTE 2.) May resume processing. Process should be recoverable without degradation.
7	STANDBY	Equipment detects fault condition.	UNSCHEDULED DOWNTIME	None	Alarm or exception report generated by same trigger.
8	UNSCHEDULED DOWNTIME	All fault conditions have been cleared.	STANDBY	Monitor all production criteria.	Transition 8 can only follow Transition 7 and may be disabled by the user. (See NOTE 3.)
9	STANDBY	A monitored para-meter has reached a pre-defined limit.	SCHEDULED DOWNTIME	None	Preventive maintenance may require reset of the monitored parameter.
10	(Any ARAMS state/substate)	User selects a new ARAMS Substate Code.	(Based on state/substate selected by user)	Depends upon state/substate selected.	The user may select a new ARAMS state or substate at any time.
11	(Any of the six basic states)	Powerdown	UNSCHEDULED DOWNTIME, SCHEDULED DOWNTIME, or NON-SCHEDULED TIME.	None	On powerup, the equipment assumes this transition has occurred. This state transition does not represent a collection event.
12	STANDBY	Equipment detects a change in standby conditions.	STANDBY	None	Transitions 12 and 13 are optional and may be disabled by the user. (See NOTE 4.)
13	PRODUCTIVE	Equipment detects a change in productive conditions.	PRODUCTIVE	None	Transitions 12 and 13 are optional and may be disabled by the user. (See NOTE 4.)
14	ENGINEERING	Equipment detects a fault condition.	UNSCHEDULED DOWNTIME	None	Transitions 14 and 15 may be disabled by the user. (See NOTE 5.)
15	UNSCHEDULED DOWNTIME	All fault conditions have been cleared.	ENGINEERING	None	Transition 14 shall only follow Transition 15. (See NOTE 6.)

NOTE 1: See the variable ARAMSState in Section 11.2.4.

NOTE 2: See PrdRecovery variable in Section 11.4.5.

NOTE 3: See SbyRecovery variable in Section 11.4.6.

NOTE 4: See SubstateSelect variable in Section 11.4.7.

NOTE 5: See EngInterrupt variable in Section 11.4.1.

NOTE 6: EngRecovery variable in Section 11.4.2 required for automatic recovery.

8.3 *General Equipment Operations Model* — The equipment has various state models that it maintains in addition to the ARAMS State Model. The state model presented in this section (Figure 3) is provided as a high-level example of one such model to clarify the relationship between ARAMS and the equipment's operations. For purposes of ARAMS, it is referred to as the General Equipment Operations Model.

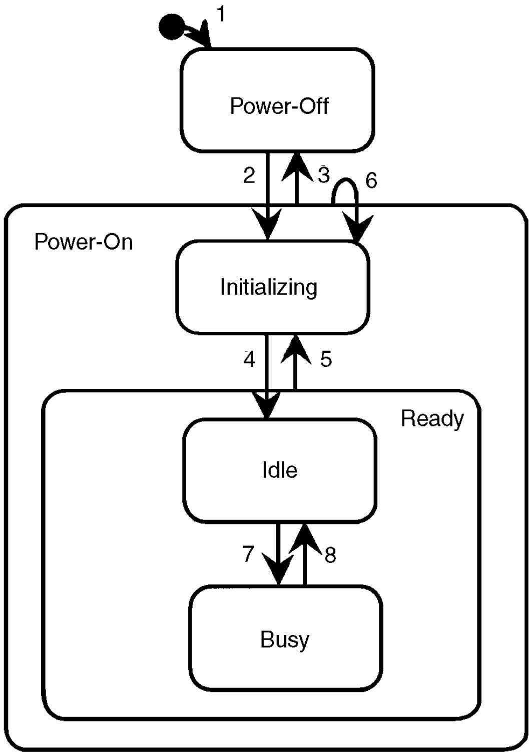

Figure 3
Example of General Equipment Operations Model

1.14.6 ARAMS assumes that a similar model, either formal or implicit, exists, although it may differ in detail depending upon the equipment type and the application. Formal implementation of the General Equipment Operations Model is not required for ARAMS compliance.

1.14.6.1 The states of the General Equipment Operations Model are not SEMI E10 states or substates.

NOTE 12: To minimize confusion between references to states and events, the events of losing power and recovering power are termed *powerdown* and *powerup* respectively.

8.3.1 *State Definitions* — The General Equipment Operations Model uses the following states:

8.3.1.1 *POWER-OFF* — The equipment has no power and is unable to function. While this state is only valid from a view external to the equipment, it is required to cover all periods of time to match the ARAMS model.

8.3.1.2 *POWER-ON* — The POWER-ON state includes all the functions of the equipment. It has two substates, INITIALIZING and READY.

8.3.1.3 *INITIALIZING* — When equipment is first powered on or recovers from a reset, it must initialize its hardware and software subsystems and components, including its different internal state models. This process takes time, which is represented by the INITIALIZING state.

8.3.1.4 *READY* — When all initializations are complete, the equipment enters the READY state and is able to interact with the user. The READY state has two substates, IDLE and BUSY.

8.3.1.5 *IDLE* — In the IDLE state, the equipment is inactive and able to accept a command for automatic processing or manual operations.

NOTE 13: IDLE is not the same as the SEMI E10 STANDBY state. (See Section 8.5.3.)

8.3.1.6 *BUSY* — In the BUSY state, the equipment is active. BUSY includes all operations of the equipment.

NOTE 14: BUSY is not the same as the SEMI E10 PRODUCTION state. (See Section 8.5.3.)

8.3.2 *General Equipment Operations Model Table of Transitions* — Table 2 defines the triggers for each transition shown in Figure 3. The second column, labeled Table 1, shows the corresponding transition in the ARAMS state model.

Table 2 Table 2 Table of Transitions for General Equipment Operations Model

#	*Table 1*	*Current State*	*Trigger*	*New State*	*Action(s)*	*Comment*
1	None	(undefined)	Equipment is installed.	POWER-OFF	None	New equipment
2	None	POWER-OFF	Power is turned on (powerup).	INITIALIZING	Begin system initialization.	Initialization of hardware and software.
3	11	POWER-ON	Power is turned off (powerdown).	POWER-OFF	None	Equipment does not function.

#	Table 1	Current State	Trigger	New State	Action(s)	Comment
4	1	INITIALIZING	System initialization is complete.	IDLE	Begin normal internal activities. Wait for input.	Normal. Equipment is able to interact with the user.
5	11	READY(any substate)	Internal error, or soft re-boot by request.	INITIALIZING	Re-initialize system.	This is treated in the same way as Transition 2.
6	1	(any substate of) POWER-ON	System reset.	INITIALIZING	Re-initialize system.	A reset is done through hardware.
7	Conditional	IDLE	Equipment receives instructions to perform an automated or manual function.	BUSY (See NOTE 1.)	Perform the requested function.	Equipment may be capable of performing multiple functions while in BUSY.
8	Continued Conditional	BUSY (See NOTE 1.)	The equipment has completed all requested functions.	IDLE	Wait for new instructions.	Equipment may be capable of performing non-hardware-related functions in IDLE.

NOTE 1: The PROCESSING ACTIVE state of the Processing State Model example in SEMI E30 would be a substate of BUSY. Other substates may exist for maintenance and diagnostics.

8.4 *State Model Relationships* — Figure 4 shows the ARAMS State Model and the General Equipment Operations Model as both simultaneously active. Certain relationships exist between the two models.

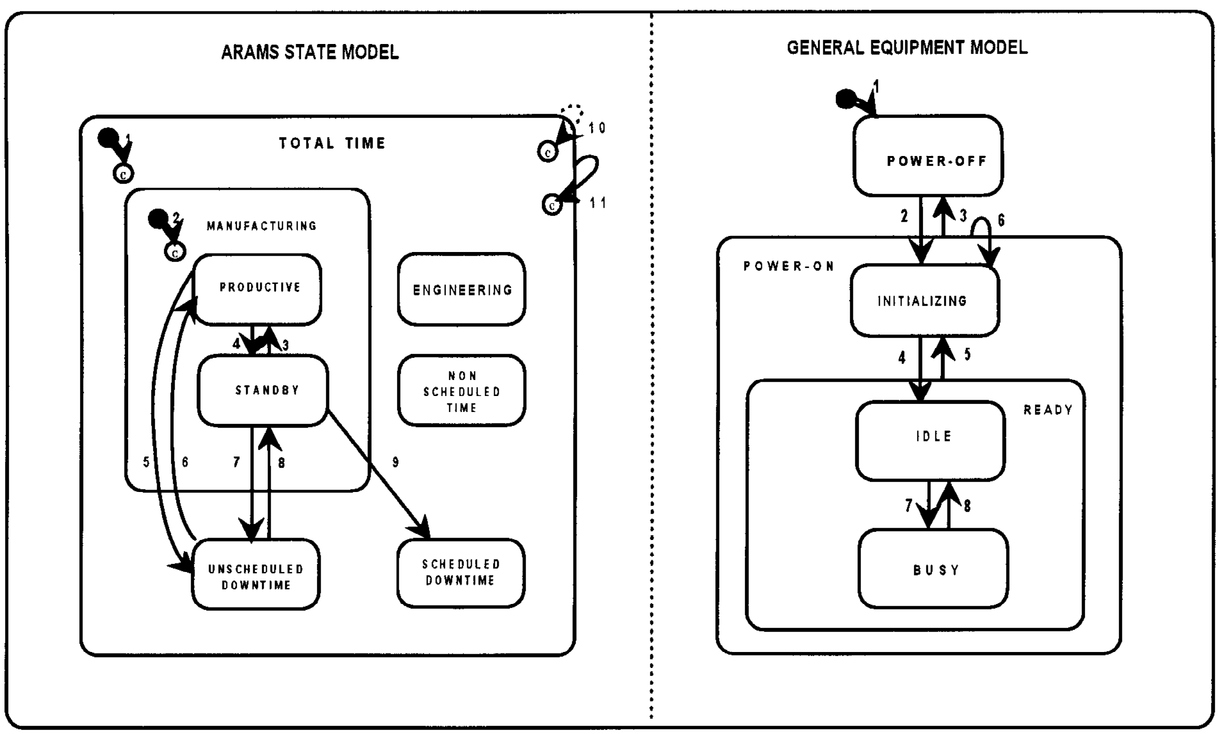

Figure 4
ARAMS State Model and General Equipment Operations Model

1.14.7 Both models cover all time, twenty-four hours each day. The ARAMS State Model describes how the equipment is used in the factory. The General Equipment Operations Model describes the equipment's activities, including the time when it has no power and is completely inactive.

1.14.7.1 The General Equipment Operations Model is provided to clarify two specific areas of relationship:

- The transitions related to powerup (#2), powerdown (#3), soft re-boot (#5), and reset (#6).
- The IDLE and BUSY activity states.

1.14.7.2 The term "reset" is used in this document to collectively represent both the soft re-boot and the hardware reset where no powerdown occurs.

1.14.7.3 Requirements related to powerup, powerdown, re-boot, and reset are specified in Section 16.

NOTE 15: Although the equipment is unable to detect a condition of no power, it is able to detect when the INITIALIZING state has been entered and is able to differentiate between a hard reset (Transition 6) and a soft re-boot (Transition 5).

8.4.1 *IDLE and BUSY* — By definition, the equipment is not in the IDLE state and in the PRODUCTIVE state at the same time, and it is not intended to be in the BUSY state while in STANDBY. The PRODUCTIVE and STANDBY states correspond to implicit substates of the BUSY and IDLE states only for the time the equipment is within the MANUFACTURING superstate — that is, it is scheduled for manufacturing (non-engineering factory operations) and is in a condition to perform its intended function.

1.14.7.4 Occasional exceptions may occur due to improper timing of a user request to change to a manufacturing state before the equipment completes a non-processing activity started in another state. For example, the equipment may be performing an automatic calibration during routine maintenance in a downtime state. In this case, the equipment determines that its production criteria are not satisfied and shall transition to STANDBY, as it is busy performing a function other than its intended function.

8.4.2 *Non-Manufacturing States* — The equipment may be in any of the POWER-OFF, INITIALIZING, IDLE, or BUSY states in the General Equipment Operations Model while in any of the non-manufacturing states: ENGINEERING, UNSCHEDULED DOWNTIME, SCHEDULED DOWNTIME, or NON-SCHEDULED TIME. For example, it may be powered off while being installed (NON-SCHEDULED TIME) or repaired (SCHEDULED or UNSCHEDULED DOWNTIME). It may be required to perform its normal processing cycle (its intended function) in ENGINEERING, NON-SCHEDULED TIME or in a downtime state.

9 ARAMS Substate Codes

1.15 This section defines the format for ARAMS Substate Codes and a set of reserved values for generic substates based on the descriptions in SEMI E10. The format is defined specifically to allow further resolution of the six basic SEMI E10 equipment states by both the factory and equipment supplier.

1.15.1 An ARAMS Substate Code consists of four ordered alphanumeric text characters, where the first two characters are reserved digits. The first character indicates the primary ARAMS state as follows:

1. PRODUCTIVE
2. STANDBY
3. ENGINEERING
4. SCHEDULED DOWNTIME
5. UNSCHEDULED DOWNTIME
6. NON-SCHEDULED TIME

1.15.2 The second character of the code, if non-zero, indicates a substate of the primary state. A zero in the second position indicates no substate has been selected. This may also be referred to as a substate of "default".

1.15.3 Codes of the form "n000" indicate no substates of a basic state have been selected. This is the "default code" for that basic state. For example, a code of "1000" indicates the PRODUCTIVE state with no substates.

1.15.4 Descriptive formatted text is defined to correspond to each ARAMS code. The first three characters of this text represent the basic state. If the next character is a forward slash "/", then subsequent text represents a substate of the basic state.

9.1 *Reserved Codes* — States and substates referenced by the reserved text in this section are based on SEMI E10 specifications. For further definition of terminology, see SEMI E10.

1.15.5 The following ARAMS Substate Codes, and the corresponding descriptive text strings for the English language (delimited by quotes), are reserved. Descriptive text with a substate of "Reserved" indicates the corresponding code is reserved for future expansion by this standard.

1.15.6 In the future, text strings in other languages may also be reserved. Equipment supporting other languages shall provide a method for the user to define alternative text strings. This provides the consistency, across different equipment, that is important to the user.

PRODUCTIVE

1000 "PRD" (default productive code)

1100 "PRD/Regular production"

1200 "PRD/Work for third parties"

1300 "PRD/Rework"

1400 "PRD/Engineering runs"

1500 "PRD/Reserved*"

1600 "PRD/Reserved"

1700 "PRD/Reserved"

1800 "PRD/Reserved"

1900 "PRD/Reserved"

STANDBY

2000 "SBY" (default standby code)

2100 "SBY/No operator"

2200 "SBY/No product"

2300 "SBY/No support tool"*

2400 "SBY/Associated cluster module down"

2500 "SBY/No host"

2600 "SBY/Reserved"

2700 "SBY/Reserved"

2800 "SBY/Reserved"

2900 "SBY/Reserved"

* NOTE: A support tool is a mechanical device used by, but not part of, the equipment. This includes cassettes, probe cards, etc.

ENGINEERING

3000 "ENG" (default engineering code)

3100 "ENG/Process experiments"

3200 "ENG/Equipment experiments"

3300 "ENG/Reserved"

3400 "ENG/Reserved"

3500 "ENG/Reserved"

3600 "ENG/Reserved"

3700 "ENG/Reserved"

3800 "ENG/Reserved"

3900 "ENG/Reserved"

SCHEDULED DOWNTIME

4000 "SDT" (default scheduled downtime code)

4100 "SDT/User maintenance delay"

4200 "SDT/Supplier maintenance delay"

4300 "SDT/Preventive maintenance"

4400 "SDT/Change of consumables"

4500 "SDT/Setup"

4600 "SDT/Production test"

4700 "SDT/Facilities-related"

4800 "SDT/Reserved"

4900 "SDT/Reserved"

UNSCHEDULED DOWNTIME

5000 "UDT" (default unscheduled downtime code)

5100 "UDT/User maintenance delay"

5200 "UDT/Supplier maintenance delay"

5300 "UDT/Repair"

5400 "UDT/Out-of-spec input material"

5500 "UDT/Change of consumables"

5600 "UDT/Facilities-related"

5700 "UDT/Reserved"

5800 "UDT/Reserved"

5900 "UDT/Reserved"

NON-SCHEDULED TIME

6000 "NST" (default non-scheduled downtime code)

6100 "NST/Unworked shifts"

6200 "NST/Equipment installation"

6300 "NST/Equipment modifications" (modify, rebuild, upgrade)

6400 "NST/Off-line training"

6500 "NST/Shutdown/startup"

6600 "NST/Reserved"

6700 "NST/Reserved"

6800 "NST/Reserved"

6900 "NST/Reserved"

9.2 *Additional Codes* — Additional codes may be defined by both the user and supplier, subject to the following constraints:

- The new code defines a refinement of a primary ARAMS state, as defined in Section 9, through use of the characters in the third and fourth positions.
- Alphabetic characters are permitted in the third and fourth positions. For purposes of sorting, these characters are assumed to be case-sensitive. All characters other than alphanumeric are prohibited.
- The third character is used to differentiate between codes defined by the user (factory) and those defined by the equipment supplier. If the third character is a digit, then the code is user defined. Otherwise, the code is supplier-defined. The user is free to assign values between "01" and "9z" as the third and fourth characters, while the supplier may assign values between "A0" and "zz".

* NOTE: Additional reserved codes may be added to Section 9.2 in the future.

1.15.7 Code definitions are exchanged as ARAMS Substate Tables, described in Section 10.3.

9.3 *Valid ARAMS Substate Code* — A valid ARAMS Substate Code is defined as any code with four alphanumeric characters where the first character is a digit between 1 and 6 and the second character is a digit between 0 and 9.

9.4 *Manufacturing Code* — A user request for the equipment to go to manufacturing specifies a special code of "0000". The code "0000" is not itself an ARAMS Substate Code and shall not be used as a code representing the current ARAMS state/substate in the variable ARAMSState.

10 ARAMS Tables

1.16 The information in this section is not required unless the equipment supports one or both of the two **ARAMS Tables** defined in Sections 10.3 and 10.4 respectively.

1.16.1 ARAMS defines two sets of data that are to be exchanged between equipment and host. These sets of data are transferred as "tables". A table is a vehicle for exchanging information and is independent of actual storage mechanisms.

1.16.2 This section introduces the concept and definitions of a generic table and defines the two specific types of tables required by ARAMS, the ARAMS Substate Table, and the ARAMS Symptom Table.

10.1 *Definition of Tables* — A table represents a general way of exchanging sets of data arranged in a tabular format. A table consists of one or more ordered sets of data, called rows, where the format and interpretation of each element of data within a row depends upon its relative position within the row, called column. Tables are transferred by providing:

- An ordered list of predefined text strings, called column headers, that identify the data element at the corresponding column position within each row,
- one or more rows of data, where each row is an ordered set of individual data elements, presented in the order specified by the column headers, and
- a set of information (attributes) about the table as a whole.

10.1.1 *Table Types and Identifier*s — Each instance of a table has a formally defined table type and a table identifier. This allows definition of general-purpose services for exchanging tables. It also allows multiple instances of a specific type of table to be referenced. OSS-compliant applications consider a table as a type of object, and a table type as a specialization of a table.

1.16.2.1 The table type definition includes specification of a reserved text string that begins with the string "Table". Table services (Section 15) and OSS services (SEMI E39) use the reserved text string as the ObjType attribute of the table object. The general table definition does not specify requirements for assigning identifiers to individual tables.

1.16.2.2 The table identifier is a text string that conforms to the requirements for an object identifier (ObjID) as specified in SEMI E39, which prohibits specific characters in ASCII. The identifier is used to identify a specific instance of a given table type and shall be unique for all tables of a given type.

1.16.2.3 A column in a table refers to all data elements at a given position across all rows of the table. Column headers are pre-defined text strings that identify the individual elements used and their relative order within each row.

1.16.3 *Table Row Definition* — A formal table type requires definition of the individual data elements within a row of the table and the specification of the column header. This information is provided in Table 3.

Table 3 Table 3 Definition for Table Row Format

Column Header	*Data Element Definition*	*Form*

Text string for column position.	Definition of data element at the corresponding column position.	Valid form for data element.

1.16.3.1 The order of elements within a row shall be invariant for a given table instance and shall strictly conform to the order in which column headers are presented. The length of any given row (the number of elements within the row) may not exceed the number of defined positions (columns). Partial rows that omit elements at the end of the row are permitted by the general definition, so long as they retain the meaning of the column positions included. That is, if a complete row consists of n columns, then an individual row may consist of the first m elements, for m ≤ n.

1.16.3.2 The first element (first column position) of a row must contain a value that is unique for all rows within a specific table instance. This value may be used as a key to identify a specific row within the table and therefore shall be a single item (e.g., not a list or structure) and may not be a floating point number. Other elements within the row may be simple lists or simple structures that do not themselves contain embedded (nested) lists or structures.

10.1.2 *Table Attributes* — It is also important to be able to exchange information about the table itself. In addition to the definition of data elements and columns, tables also have predefined attributes. The general table definition specifies three attributes: the number of columns, the number of rows, and the size of the table exclusive of formatting used for storing or transferring the table. (See Section 13.2.)

10.1.3 *Additional Requirements* — Specific table types may have additional requirements and restrictions.

1.16.3.3 ARAMS tables, defined in Sections 10.3 and 10.4, have additional requirements as follows:

- Tables shall be stored in non-volatile memory and shall be capable of subsequent modifications.
- Elements within a row shall follow the order specified in the table definitions.
- All rows shall consist of the full number of columns specified. Partial rows are prohibited.

1.16.3.4 A table is considered structurally valid if it conforms to the requirements in this section and to the row format defined for its specific type. A structurally valid table may or may not have correct content — that is, the values within any given row may or may not meet other requirements or expectations.

1.16.3.5 ARAMS tables are provided only for use by the operator and the associated data sent to the host. Equipment shall not reject a user-defined table that is structurally valid for its table type because of its contents. Equipment shall not use the contents of tables to validate ARAMS Substate Codes or symptom information provided by the host.

1.16.3.6 Certain errors in the contents may cause the tables to not work properly in some way. For example, if a row identifier is used more than once, then the equipment may fail to find any but the first occurrence. However, the user shall accept responsibilities for any errors in the contents of ARAMS tables.

10.2 *ARAMS Substate Table* — The ARAMS Substate Table provides extensions to the generic substate definitions contained in Section 9.

1.16.4 Each row in an ARAMS Substate Table consists of a four-character ARAMS Substate Code as the first element of the row, and a corresponding text string containing a brief description of the state as the second element. Table 4 defines the ARAMS Substate Table row format.

Table 4 Table 4 ARAMS Substate Table Row Definition

Column Header	*Data Element Definition*	*Form*
"Code"	ARAMS Substate Code.	Four-character text string.
"Text"	Brief description of ARAMS Substate Code.	Formatted text, Sections 9.1, 9.2.

1.16.5 The descriptive text is displayed as a prompt or selection item for the operator when selecting a new ARAMS state. The description selected is placed in the variable ARAMSText and the corresponding ARAMS Substate Code is placed in the variable ARAMSState as the new ARAMS state (see Section 11.2).

1.16.6 Extensions to ARAMS codes may be defined by either the user or supplier according to the rules in Section 9.3. The ARAMS Substate Table provides a formal method for each to obtain the codes and text defined by the other. The table's identifier provides a method for differentiating between the three sets of information (generic definitions, user-defined definitions, and supplier-defined definitions) when necessary to remove ambiguity in communications. The user at all times shall be able to select an ARAMS state or substate based on the combination of the three sources of definitions.

1.16.7 An ARAMS Substate Table has a table type of "TableARAMSCode". The row format is defined in Table 4.

10.3 *ARAMS Symptom Table* — The user may define an optional table of common human-observable symptoms called the ARAMS Symptom Table, with a table type of "TableARAMSSymptom". The row of this table is defined in Table 5 below.

Table 5 Table 5 ARAMS Symptom Table Row Definition

Column Header	*Data Element Definition*	*Form*
"ID"	Identifier of ARAMS symptom.	Unsigned integer.
"Indicator"	First one to four characters of ARAMS code to which symptom may be applied.	Formatted text. 1 to 4 characters. (See Sections 9.2–9.3.)
"Text"	Brief description of ARAMS symptom.	Text. Maximum length is 80 characters.

1.16.8 Each table row consists of three items: a symptom identifier, an ARAMS substate indicator, and a text string describing the symptom.

1.16.9 The symptom identifier and text are analogous to the equipment's alarm or exception identifier and corresponding text description. The identifier and text should each be unique. A symptom identifier with a value of zero is reserved to indicate "no symptom".

1.16.10 The substate indicator consists of the first one to four characters of an ARAMS Substate Code and indicates the ARAMS state or state/substate where the symptom is to be applied. For example, a symptom that is to be applied whenever the user requests a change to the UNSCHEDULED DOWNTIME state, including any of its substates, is assigned an indicator of "5". A symptom that is to be applied in any case of "UDT/change of consumable" would use an indicator of "55" as the first two digits of the corresponding ARAMS code.

1.16.11 Whenever the operator requests a change in the ARAMS state or substate, the operator shall be presented with the text descriptions from the appropriate Symptom Table. The text description of the selection is placed in the status variable SymptomText, and the corresponding identifier is placed in SymptomID.

11 ARAMS Data

1.17 This section defines the data requirements for ARAMS. This includes requirements for retaining data across a powerdown or reset. Data retention requirements are classified for each item of ARAMS data in one of the following ways:

Class 1 — A "soft" value that is reset to a default value as part of system initialization (e.g., DowntimeAlarm)

Class 2 — A "hard" value that never changes (e.g., "EqpSerialNum")

Class 3 — A "firm" value that changes infrequently and is to be retained in non-volatile memory (e.g., "EqpName")

Class 4 — A "dynamic" value that is retained in non-volatile memory and changes whenever an ARAMS state transition occurs.

Class 5 — A "very dynamic" value that may change rapidly and is important to retain but does not require absolute accuracy (e.g., CycleCtr for high throughput equipment). Current values shall be saved in non-volatile memory periodically at least once per minute.

1.17.1 Except where otherwise specified, data is assumed to be Class 1.

1.17.2 Variables within each section are presented in alphabetical order.

11.1 *Status Information* — Status data is information that is maintained by equipment and in general cannot be changed by the host. This section defines specific elements of status data that the user is able to read upon request and that may be included in event report messages to the host.

1.17.3 A mechanism shall be provided for the user to read these values.

1.17.4 All date/time values are text strings with a timestamp format (see Section 4).

NOTE 16: The timestamp format is not a requirement for clock precision.

NOTE 17: Subsequent references to subsections of 11.2 below are using the original numbers.

11.1.1 *ARAMSInfo* — Additional information set or cleared by the equipment at the time an ARAMS state transition occurs. Optional. Form: text. 0–80 characters.

11.1.2 *ARAMSState* — The ARAMS Substate Code corresponding to the ARAMS state/substate that became active following the most recent ARAMS state transition. When an ARAMS state transition occurs, ARAMSState is the specific code representing the new ARAMS state/substate. It shall be retained as Class 4 data. The value of "0000" is prohibited. Form: formatted alphanumeric text. 4 characters.

11.1.3 *ARAMSText* — The descriptive text corresponding to the code in ARAMSState. Form: text. 5–80 characters.

11.1.4 *Clock* — Contains the current value of the date and time at the equipment. When included in an event report, Clock represents the timestamp for the occurrence of the event. Clock may represent the current value of a real-time date/time clock available upon demand. Form: formatted text.

11.1.5 *CycleCtr* — The number of cycles (equipment cycles) during the lifetime of the equipment (not resettable). It shall be retained as Class 5 data. Form: unsigned integer.

NOTE 18: See SEMI E10 for a formal definition of cycle.

11.1.6 *DowntimeAlarm* — Identifier of the most recent alarm or exception triggering an equipment-initiated transition to SCHEDULED DOWNTIME or UNSCHEDULED DOWNTIME from the PRODUCTIVE or STANDBY states. This value is reset to zero for all other ARAMS state transitions to indicate "no associated alarm/exception". Form: unsigned integer.

11.1.7 *DowntimeAlarmText* — Text associated with DowntimeAlarm. This value is cleared (set to a zero-length string) for all ARAMS state transitions except the transition to SCHEDULED DOWNTIME or UNSCHEDULED DOWNTIME from the PRODUCTIVE or STANDBY states. Form: text. 0–80 characters.

11.1.8 *DowntimeData* — Text associated with transitions to, or within, the SCHEDULED or UNSCHEDULED DOWNTIME states. For transitions following powerup/reset (see Section 8.5.2), the value in DowntimeData shall be set to "Power Loss". This value is reset to a zero length (empty) string for other ARAMS state transitions. When associated with equipment initiated transitions, this may be used to carry fault information (e.g., the component serial number of a repaired component, as defined by the equipment supplied). When associated with operator-initiated transitions, it may consist of comments entered at the equipment's control panel. Form: text. 0–256 characters.

11.1.9 *LastPowerdown* — Contains a date/time timestamp that estimates the time when the last loss of power or reset occurred (Transition 11), based on the value in PowerdownTime on powerup. It shall be retained as Class 3 data. Form: formatted text.

11.1.10 *PowerdownTime* — Contains a date/time timestamp used as an estimate for the timestamp for loss of power or reset and used for reports associated with ARAMS state Transition 11. It shall be retained as Class 5 data. Form: formatted text.

11.1.11 *PrdState* — The ARAMS code corresponding to the last user-specified ARAMS Substate Code for PRODUCTIVE. Used as the default substate for PRODUCTIVE. Initialized to "1000" at Transition 1. Form: formatted alphanumeric text. 4 characters.

11.1.12 *PrevARAMSState* — The ARAMS code corresponding to the ARAMS state/substate that was active immediately preceding the most recent ARAMS state transition. This value is useful for resynchronization after a period of inability to communicate. It shall be retained as Class 4 data. Form: formatted alphanumeric text. 4 characters.

11.1.13 *SymptomID* — A numeric identifier of a symptom selected by the user when requesting the equipment to change states. A value of zero is equivalent to a selection of "no symptom". Form: unsigned integer.

11.1.14 *SymptomText* — The descriptive text for the symptom selected by the user when requesting the equipment to change states. A zero-length string corresponds to a selection of "no symptom". Form: text. 0–80 characters.

11.2 *Constant Data* — Equipment shall provide and maintain the following values as class 2 data. These values are set by the equipment manufacturer and shall not be changeable by the user. These data elements are required for equipment with formally defined components that are also ARAMS compliant, such as clustertools. This data is used for event reports, to identify the original source of the event.

11.2.1 *EqpModel* — A text string containing the equipment model. Form: 1–80 characters.

11.2.2 *EqpSerialNum* — A text string containing the product serial number assigned by the supplier. Form: 1–80 characters.

11.3 *User-Configurable Data* — The data elements in this section shall be settable by both the operator and the host. They shall be retained as Class 3 data.

11.3.1 *EngInterrupt (optional)* — A Boolean value that enables (TRUE) or disables (FALSE) equipment-initiated transitions from ENGINEERING to UNSCHEDULED DOWNTIME (Transition 14 in Figure 2) when a fault condition is detected. This variable is required if EngRecovery is supported. Form: Boolean.

11.3.2 *EngRecovery (optional)* — A Boolean value that enables (TRUE) or disables (FALSE) automatic recovery returning to ENGINEERING from UNSCHEDULED DOWNTIME (Transition 15 in Figure 2) following an equipment-detected fault that has spontaneously cleared. Form: Boolean.

11.3.3 *EqpName (required)* — A text string representing the user-assigned logical name of the equipment. Form: 1–80 characters.

NOTE 19: For OSS-compliant equipment, this corresponds to the ObjID attribute of the object “Agent”, “PM”, “CM”, or “TM” in other standards.

11.3.4 *PowerupState (optional)* — A text character indicating the powerup (default entry) state when a powerdown occurs during manufacturing time. May be either “2” for STANDBY or “5” for UNSCHEDULED DOWNTIME.

11.3.5 *PrdRecovery (optional)* — A boolean value that enables (TRUE) or disables (FALSE) the equipment-initiated return to PRODUCTIVE (Transition 6 in Figure 2) following an equipment-detected fault that has cleared without operator intervention. Form: boolean.

11.3.6 *SbyRecovery (optional)* — A boolean value that enables (TRUE) or disables (FALSE) the equipment-initiated return to STANDBY (Transition 8 in Figure 2) following an equipment-detected fault that has cleared. Form: boolean.

11.3.7 *SubstateSelect (optional)* — A Boolean value that enables (TRUE) or disables (FALSE) equipment-initiated selection of substates of STANDBY and PRODUCTIVE (including Transitions 12 and 13). Form: Boolean.

11.4 *Accumulators* — This section defines a set of optional accumulators that either count time in one of the six basic SEMI E10 states or count interruptions occurring in the PRODUCTIVE state. It shall be possible for both the operator and host to (1) adjust the values in the individual accumulators, and (2) reset all seven values to zero simultaneously. Operator and host assume responsibility for the edited values and their accuracy. Equipment shall not impose restrictions on the modified values.

1.17.5 In addition, two timestamps are required for support of the accumulators. These timestamps shall not be changed by the user.

1.17.5.1 Where accumulators are supported, the variables defined in this section shall be supported as a set as Class 4 data.

1.17.5.2 The time accumulators are reported in minutes with an accuracy of ± one minute. Internal representations of the accumulators in smaller units may be required for short processes.

11.4.1 *ARAMSAccumReset* — The timestamp of when the following seven accumulators were last reset to zero. Form: formatted numeric text.

11.4.2 *ARAMSTimestamp* — The timestamp of the most recent ARAMS state change to the state identified in ARAMSState. This value is updated whenever a transition to a new ARAMS state occurs and is used in conjunction with PowerdownTime to update the accumulators following a powerup or reset. Form: formatted numeric text.

11.4.3 *EngTime* — Time in ENGINEERING.

11.4.4 *InterruptionPrd* — The number of transitions to UNSCHEDULED DOWNTIME from PRODUCTIVE.

NOTE 20: This includes the equipment-initiated Transition 5, a user-initiated transition covered by Transition 10, and the powerdown Transition 11, when occurring from PRODUCTIVE.

11.4.5 *InterruptionTotal* — The total number of transitions to UNSCHEDULED DOWNTIME from any state. This includes equipment-initiated transitions and user-initiated transitions.

11.4.6 *NSTime* — Time in NON-SCHEDULED TIME.

11.4.7 *PrdTime* — Time in PRODUCTIVE.

11.4.8 *SbyTime* — Time in STANDBY.

11.4.9 *SDTime* — Time in SCHEDULED DOWNTIME.

11.4.10 *UDTime* — Time in UNSCHEDULED DOWNTIME.

12 Events and Pre-Defined Event Reports

1.18 Each transition in the ARAMS State Model, with the exception of Transitions 2 and 11, represents a unique collection event that shall be reportable to the host. A generic collection event ARAMS State Change shall be provided that corresponds to any ARAMS state change. This allows greater efficiency for common data to be placed in a single report.

1.19 The equipment shall support one of the two following requirements:

- The capability of allowing the host to define reports associated with each ARAMS state change event, or
- A set of pre-defined (default) event reports for the ARAMS State Change Event, equipment-initiated transitions to UNSCHEDULED DOWNTIME (Transitions 5 and 7, and Transition 1 as appropriate), and user-initiated transitions to UNSCHEDULED DOWNTIME.

1.20 The pre-defined reports, defined below, are labeled "A", "B", and "C", for ease of reference only. Report "A" represents the information required for every ARAMS State Change Event. Implementations of these reports shall provide, at a minimum, the updated values of the indicated variables.

Report "A"

ARAMS State Change Event

EqpModel

EqpSerialNum

EqpName

Clock

ARAMSState

PrevARAMSState

Report "B"

Equipment-Initiated Transitions to UNSCHEDULED DOWNTIME Event

DowntimeAlarm

DowntimeAlarmText

DowntimeData

Report "C"

User-Initiated Transition

SymptomID

SymptomText

DowntimeData

13 Object Services Compliance

This section provides information required for equipment that is OSS-compliant. OSS compliance is not required for ARAMS compliance.

13.1 *Equipment Object* — ARAMS-compliant equipment (equipment subsystems) that are compliant to OSS (SEMI E39) and that provide an object representing the equipment (equipment subsystems) shall include as attributes of that object the attributes listed in Table 6.

1.20.1 The data element Clock may be represented by a Clock object. In this case, Clock is not an attribute of the Equipment object but is a component of the Equipment object and is accessible through OSS services.

1.20.2 Table 6 defines the variables specified in earlier sections as object attributes conformant with the requirements of OSS. The column labeled "Access" indicates whether the attribute is read-only (RO) or read-write (RW), and the column labeled "Reqd" indicates if the attribute is required for ARAMS compliance.

Table 6 Table 6 ARAMS Object Attribute Definitions

Attribute Name	*Definition*	*Access*	*Reqd*	*Form*
ARAMSAccumReset	Timestamp for when time-in-state accumulators were last reset to zero. Required if accumulators are supported.	RO	N	Formatted numeric text: timestamp format.
ARAMSInfo	Additional information set by the equipment at the time of an ARAMS state transition.	RO	N	Text. 0–80 characters.
ARAMSState	Code corresponding to currently selected ARAMS state/substate.	RO	Y	Formatted alphanumeric text. 4 characters.
ARAMSText	Descriptive text corresponding to code in ARAMSState.	RO	Y	Text. 5–80 characters.

Attribute Name	*Definition*	*Access*	*Reqd*	*Form*
ARAMSTimestamp	Timestamp of last ARAMS state change.	RO	N	Formatted numeric text: timestamp format.
Clock (See NOTE 1.)	Current time and date.	RW	Y	Formatted numeric text: timestamp format.
CycleCtr	Non-resettable counter of equipment run cycles.	RO	Y	Unsigned integer.
DowntimeAlarm	Identifier of last alarm or exception triggering an equipment-initiated transition to a downtime state and/or substate.	RO	Y	Text. Conforms to ObjID.
DowntimeAlarmText	Text associated with DowntimeAlarm.	RO	Y	Text. 0–80 characters.
DowntimeData	Equipment-defined text associated with transitions to downtime state and/or substate.	RO	Y	Text. 0–256 characters.
EngInterrupt	Enables (TRUE) or disables (FALSE) an equipment-initiated transition from ENGINEERING to UNSCHEDULED DOWNTIME. Required in EngRecovery is supported.	RW	N	Boolean.
EngRecovery	Enables (TRUE) or disables (FALSE) automatic recovery from UNSCHEDULED DOWNTIME to ENGINEERING.	RW	N	Boolean.
EngTime	Accumulated minutes in the ENGINEERING state/substate. Required if accumulators are supported.	RW	N	Unsigned integer.
EqpModel	The equipment model.	RO	Y	Text. 1–80 characters.
EqpSerialNum	Equipment serial number.	RO	Y	Text. 1–80 characters.
InterruptionPrd	Counts number of transitions from PRODUCTIVE to UNSCHEDULED DOWNTIME. Required if accumulators are supported.	RW	N	Unsigned integer.
InterruptionTotal	Counts total number of transitions to UNSCHEDULED DOWNTIME. Required if accumulators are supported.	RW	N	Unsigned integer.
LastPowerdown	Estimate of last powerdown time.	RO	Y	Formatted numeric text: timestamp format.
NSTime	Accumulated minutes in NON-SCHEDULED TIME state/substate. Required if accumulators are supported.	RW	N	Unsigned integer.
PowerdownTime	Estimate of powerdown time.	RO	Y	Formatted numeric text: timestamp format.
PowerupState	Indicates the powerdown state when powerdown occurs during manufacturing time.	RW	N	Text digit: “2” = standby, “5” =unscheduled downtime.
PrdRecovery	Enables (TRUE) or disables (FALSE) the equipment-initiated return to PRODUCTIVE from UNSCHEDULED DOWNTIME. Required for support of automatic equipment-initiated recovery.	RW	N	Boolean.
PrdState	The last ARAMS code specified by the user for PRODUCTIVE.	RO	Y	Formatted alphanumeric text. 4 characters.
PrdTime	Accumulated minutes in PRODUCTIVE state/substate. Required if accumulators are supported.	RW	N	Unsigned integer.
PrevARAMSState	ARAMS Substate Code corresponding to state prior to last state transition.	RO	Y	Formatted alphanumeric text. 4 characters.
SDTime	Accumulated minutes in SCHEDULED DOWNTIME state/substate. Required if accumulators are supported.	RW	N	Unsigned integer.
SbyRecovery	Enables or disables the equipment-initiated return to STANDBY from UNSCHEDULED DOWNTIME. Required for support of automatic equipment-initiated recovery.	RW	N	Boolean.
SbyTime	Accumulated minutes in STANDBY state/substate. Required if accumulators are supported.	RW	N	Unsigned integer.

Attribute Name	Definition	Access	Reqd	Form
SubstateSelect	Enables (TRUE) or disables (FALSE) equipment-initiated selection of substates in PRODUCTIVE and STANDBY.	RW	N	Boolean.
SymptomID	The identifier of a symptom selected by the user when requesting a state change.	RO	Y	Text. Conforms to ObjID.
SymptomText	The descriptive text of the symptom selected by the user when requesting a state change.	RO	Y	Text. 0–80 characters.
UDTime	Accumulated minutes in UNSCHEDULED DOWNTIME state/substate. Required if accumulators are supported.	RW	N	Unsigned integer.

NOTE 1: The Clock attribute may be replaced by a Clock object having a DateTime attribute, as defined in SEMI E38.

1.21 *Table Objects* — Table 7 defines the attributes of a generic table. This is an abstract type of object that is not itself directly implemented. Specific table types, which require definitions of their row formats (Section 10.2), are subtypes of the generic table supertype and inherit all of the attributes of the supertype (see SEMI E39, Appendix).

Table 7 Table 7 Table Object Attribute Definition

Attribute Name	Definition	Access	Reqd	Form
ObjType	The object type.	RO	Y	Text = “Table”.
ObjID	The object’s identifier.	RO	Y	Text. 1–80 characters.
NumCols	Number of columns.	RO	Y	Unsigned integer.
NumRows	Number of rows.	RO	Y	Unsigned integer.
DataLength	Total number of bytes required to store the table elements, exclusive of any formatting required for storage or transfer.	RO	Y	Unsigned integer.

1.22 Different applications may define additional table attributes for specific table types. The subtypes defined by ARAMS (Sections 10.3 and 10.4) are “TableARAMSCode” and “TableARAMSSymptom” and have no additional attributes.

2 Human Interface Requirements

This section provides a central location for requirements affecting the human interface. It specifies the functions that shall be available to the operator through the equipment’s console (human interface), where the equipment provides such an interface. Applications that do not otherwise provide such an interface (e.g., cluster modules) are exempted from these requirements.

13.2 *Data Access* — All supported data elements defined in Section 11 shall be accessible to the operator. Specifically, the operator shall be able to view all ARAMS data provided by the equipment and shall be able to modify the values of all data defined in Section 11.4.

2.1.1 Where the accumulators defined in Section 11.5 are supported, the operator shall be able to change their individual values, with the exception of ARAMSAccumReset, and to reset the entire set together.

13.3 *Selection of an ARAMS State/Substate* — The operator shall be able to select any of the standard ARAMS states and substates defined in Section 9.2, exclusive of those definitions with a “Reserved” substate that are reserved for future standards. The operator shall also be able to select MANUFACTURING.

2.1.2 Where ARAMS Substate Tables are provided with user-defined or supplier-defined extensions, these extensions shall also be available to the operator for selecting a change in the current ARAMS state and /or substate.

2.1.3 When selecting a SCHEDULED DOWNTIME or UNSCHEDULED DOWNTIME state/substate, the operator shall be able to enter a comment to be stored in DowntimeData (Section 11.2.10).

13.4 *User-Defined ARAMS Symptom Tables* — Where ARAMS Symptom Tables are supported, the descriptive text for symptoms provided within existing ARAMS Symptom Tables shall be presented to the operator for selection, according to the specifications in Section 10.4, at the time the operator selects an ARAMS State (Section 14.3). The operator is not required to select a symptom. If the operator selects a symptom, the symptom identifier is stored in the variable SymptomID and the descriptive text is stored in SymptomText.

13.5 *Table Access* — Where the equipment supports one or both types of tables specified by ARAMS (Sections 10.3 and 10.4), the operator shall have full access to entries within those tables that are user-defined. The operator shall be able to enter (define), modify, and delete user-defined entries within ARAMS table. The operator shall not be able to change or delete the generic ARAMS Substate Codes defined in Section 9 or supplier extensions to those generic codes.

2.1.4 The operator shall be able to read the contents of all existing ARAMS tables.

13.6 *Color Codes* — The use of color associated with the display of ARAMS state information is optional. To ensure consistency for the user, the following color schemes shall be applied for all cases where color is associated with an ARAMS state.

Uptime states

PRODUCTIVE: green

STANDBY: yellow

ENGINEERING: blue

Downtime states

SCHEDULED DOWNTIME: light red

UNSCHEDULED DOWNTIME: red

Other

NON-SCHEDULED TIME: grey

2.1.5 This scheme provides a visual grouping.

NOTE 21: The colors used for the two downtime states should be both clearly related and readily distinguishable, such as pink and bright red.

3 Services

3.1 This section formally defines the message services specifically required to support ARAMS functionality.

3.1.1 Formal definition has three parts:

- A list of the services defined (Table 8),
- a common parameter dictionary defining all of the parameters of these services (Table 9), and
- a dictionary of the individual services that defines the parameters used by each service.

3.1.2 Services that are used by ARAMS, but are not specific to ARAMS, are defined by other standards and are referenced in Section 3.

3.1.3 Both the equipment and the host provide the services TableSend and TableRequest, and both may use the services provided by the other. The service ARAMSStateChange is provided by equipment only.

3.1.4 Table 8 lists the ARAMS services. There are two types of services:

- An initial message and response between the service user and the service provider.
- A notification message from the service provider to the service user that does not require a response.

3.1.5 The column in Table 8 that is called “Type” is used to indicate whether the service consists of a request/response message pair (R) or a single notification message (N).

Table 8 Table 8 ARAMS Services

Message Service Name	*Type*	*Description*
TableSend	R	Used by the service user to send or delete a table.
TableRequest	R	A request to receive a table.
ResetAccumulators	R	A request to reset the set of accumulators.
ARAMSStateChange	R	Request to go to a specified ARAMS state/substate.

3.1.6 Figure 5 shows the message flow for each of the above ARAMS services.

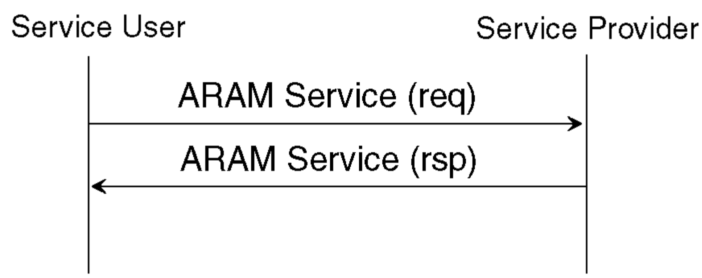

Figure 5
ARAM Service Message Flow

13.7 *Services Parameter Dictionary* — Table 9 defines all of the parameters, including the elements of complex parameters, used in ARAMS message services. Parameters are listed in alphabetical order.

3.1.7 The “Form” column in Table 9 indicates the data type for the parameter. A list of the standard data types for “Form” is included in Section 4.3.

Table 9 Table 9 ARAMS Services Parameter Dictionary

Parameter	*Definition*	*Form*
ARAMSCode	ARAMS state/substate code.	Alphanumeric text with fixed length of four characters.
AttrData	The value of an attribute.	Varies with attribute.
AttrName	The attribute’s name.	Text. Varies with object type.
ColHdr	Column header.	Text. 1–20 characters.
ErrorCode	Contains the code for the specific error found.	Enumerated.
ErrorText	Text in support of the error code to provide additional information.	Text.
ObjSpec	The object specifier, used to specify the owner of the target object.	Formatted text.
RequestStatus	The result of the request to change to a new ARAMS state/substate.	Enumerated: 0 = Acknowledge, action has been effected. 2 = Cannot perform now. 3 = Invalid parameter 4 = Acknowledge, action will be performed with completion signaled later by an event. 6 = Object unknown.
Status	Error information.	Structure consisting of ErrorCode and ErrorText.
SymptomID	The numeric symptom identifier.	Unsigned integer.
SymptomText	Descriptive text for symptom.	Text. 0–80 characters.
TableAck	Indicates the success (=T) or failure (=F) of the operation.	Boolean.
TableAttr	A table attribute other than object (table) type or object (table) identifier. (See NOTE 1.)	Structure consisting of AttrName and AttrData.
TableCmd	Instructions concerning the table transfer.	Enumerated: Entire table Add new rows Append new columns Replace existing rows Replace existing columns
TableElem	One of the basic elements of a table.	Varies with table type and (column) position within the row.
TableID	Table identifier.	Text. 1–80 characters.

Parameter	Definition	Form
TableRow	A row of the table in terms of its elements TableElem.	Structure, consisting of different table elements, following the order of the column headers.
TableStatus	Information concerning the success or failure of the operation.	Structure consisting of TableAck and (List of) Status.
TableType	Type of table.	Text = "TableARAMSCode" or "TableARAMSSymptom".

NOTE 1: ARAMS tables use only the general table attributes: number of rows, number of columns, and table length.

3.2 *TableSend* — The TableSend service is used to transfer or delete a table.

3.2.1 Table 10 defines the parameters for the TableSend service. A table that is not structurally valid shall be rejected and discarded.

Table 10 Table 10 TableSend Service

Parameter	Req/Ind	Rsp/Conf	Description
ObjSpec	C	-	The object specifier, used to specify the owner of the target object. When omitted, the receiver of the message is intended.
TableType	M	M	Signifies the type of table to be sent.
TableID	C	C(=)	Table identifier. Required for table type TableARAMSSymptom.
TableCmd	M	-	Instructions about the table being transferred.
TableAttr (list of)	M	-	The table's attributes.
ColHdr (list of)	M	-	Table column headers.
TableRow (list of)	C	-	Table contents, organized by rows. Rows may or may not be ordered, depending on the definition for a specific table type. Must be omitted when deleting a table.
TableStatus	-	M	Information concerning the success or failure of the request.

3.3 *TableRequest* — The TableRequest service is used to request the service provider to send a table. A table that is not structurally valid shall be discarded.

3.3.1 Table 11 defines the parameters for the TableRequest service.

Table 11 Table 11 TableRequest Service

Parameter	Req/Ind	Rsp/Conf	Description
ObjSpec	C	-	The object specifier, used to specify the owner of the target object. When omitted, the receiver of the message is intended.
TableType	M	M	Signifies the type of table to be sent.
TableID	C	C(=)	Table identifier. Required for table type TableARAMSSymptom.
TableCmd	M	-	Instructions about the table being transferred.
TableElem (list of)	C	-	Row identifiers of requested rows. If omitted, entire table is requested.
TableAttr (list of)	-	M	Table attributes.
ColHdr (list of)	-	M	Table column headers. Omitted only if table for specified type and identifier does not exist.
TableRow (list of)	-	M	Table rows. Omitted only if table for specified type and identifier does not exist.
TableStatus	-	M	Information concerning the success or failure of the request.

3.4 *ARAMSStateChange* — The ARAMSStateChange service is sent to the service provider to request a change to a new ARAMS state/substate.

3.4.1 Table 12 defines the parameters for the ARAMSStateChange service.

Table 12 Table 12 ARAMSStateChange Service

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
ObjSpec	C	-	The object specifier, used to specify the owner of the target object. When omitted, the receiver of the message is intended.
ARAMSCode	M	-	New ARAMS state/substate requested.
SymptomID	C	-	ARAMS Symptom ID. Optional.
SymptomText	C	-	Symptom description. Optional.
RequestStatus	-	M	Information concerning the success or failure of the request.

3.5 *ResetAccumulators* — The ResetAccumulators service is sent to the service provider to request that the set of accumulators defined in Section 11.5 be reset.

3.5.1 Table 13 defines the parameters for the ResetAccumulators service.

Table 13 Table 13 ResetAccumulators Service

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
ObjSpec	C	-	The object specifier, used to specify the owner of the target object. When omitted, the receiver of the message is intended.
RequestStatus	-	M	Information concerning the success or failure of the request.

4 ARAMS Behavioral Requirements

This section specifies the behavior required for ARAMS compliance.

13.8 *ARAMS State Transitions* — ARAMS compliance requires that the equipment be capable of Transitions 1 through 11 in Table 1, with the exception of Transition 9. Transitions 12 and higher are optional.

4.1.1 Transition 9 is required only of equipment that puts itself in a preventive maintenance mode based on one or more internally monitored parameters, such as a cycle counter. Transition 9 specifies that such a transition takes place from STANDBY to SCHEDULED DOWNTIME only. For equipment without this feature, Transition 9 never occurs.

4.1.2 The equipment is required to provide the capability to report to the host each time a state transition occurs and to include data related to that transition in these reports. Transition data included in these reports allows the factory to calculate the amounts of time spent in the different states.

4.1.3 At all transitions except Transition 10, when a manufacturing state is specified, and Transition 11, the current value of ARAMSState is stored in PrevARAMSState and the ARAMS Substate Code specified is then stored in ARAMSState. A user selection of a specific manufacturing state is discussed in Section 16.5 below.

4.1.4 For all equipment-initiated transitions, except as specified in Sections 16.5 and 16.6, the equipment shall use an ARAMS Substate Code representing the default for that state and shall not select a substate.

4.1.5 DowntimeAlarm and DowntimeAlarmText are set only for equipment-initiated transitions to SCHEDULED DOWNTIME and UNSCHEDULED DOWNTIME. For all other transitions, these variables are cleared.

4.1.6 If the equipment supports ARAMS accumulators (Section 11.5), then the time spent in the previous state is calculated and the appropriate accumulator is updated at this time.

13.9 *Powerup Entry* — The ARAMS State Model becomes active during or after initialization following a powerup or reset when Transition 1 occurs. The state entered at Transition 1 (the default entry state) depends upon the ARAMS state that was last active before the powerdown or reset occurred. This is determined by the ARAMS Substate Code stored in ARAMSState (Section 11.2.4). If this value is not a valid ARAMS Substate Code as defined in Section 9.4, then the default entry state is NON-SCHEDULED TIME.

4.1.7 At Transition 1, if the value in Powerdown-Time does not represent a valid date and time, it shall be set to a string of all zeroes. The value stored in PowerdownTime is then stored in LastPowerdown to preserve the estimate of the timestamp of the last powerdown or reset. PowerdownTime is next set to the current date and time, and periodic updating of Class 5 variables is enabled.

4.1.8 Normal communications to both operator and host are unavailable from powerdown to a point within or following system initialization. It is recommended that initializations affecting communications with the host precede initialization of the ARAMS State Model, to allow an event report to be sent to the host on Transition 1. It is highly desirable to the host to receive an event report for Transition 1. However, it is not required for ARAMS compliance.

4.1.9 Transition 11 never generates an event report.

13.10 *Powerup and Powerdown States* — The ARAMS state entered at Transition 1 is the default or powerup entry state for the ARAMS State Model. The powerdown state is the state that was active at Transition 11 and is determined by the value stored in ARAMSState at the time of Transition 1.

4.1.10 If the value in ARAMSState indicates any non-manufacturing state (ENGINEERING, SCHEDULED DOWNTIME, UNSCHEDULED DOWNTIME, or NON-SCHEDULED TIME), it is regarded by the factory as continuing in that state both during the time it is powered off and at Transition 1. The powerup entry state is the same as the powerdown state.

4.1.10.1 If the equipment is in PRODUCTIVE or STANDBY and is powered off, it is regarded by the factory as in UNSCHEDULED DOWNTIME from the time of the powerdown, through any subsequent powerup, and until it is specifically put into a different state by the user. The ARAMS powerup entry state either is UNSCHEDULED DOWNTIME or is determined by an optional variable PowerupState (Section 11.4.2) to be either UNSCHEDULED DOWNTIME or STANDBY.

4.1.10.2 If the ARAMS state prior to powerdown cannot be determined (e.g., if it has never been set or is invalid), the state entered after powerup is NON-SCHEDULED TIME.

4.1.10.3 Time-in-state calculations for the powerdown state assume that any transition to a new state occurs at the time of powerdown, re-boot, or reset.

13.10.1 *PowerupState* — PowerupState, where supported, shall be configurable by the user to specify either UNSCHEDULED DOWNTIME or STANDBY as the powerup entry state after a powerdown has occurred from PRODUCTIVE or STANDBY. The impact of loss of power has different effects on different types of equipment. An entry state of STANDBY allows those types of equipment that do not have safety or setup concerns to be powered off and on and returned to manufacturing. PowerupState contains a text character of either “2” (STANDBY) or “5” (UNSCHEDULED DOWNTIME) and has an initial default value of “5”. If the value in PowerupState is neither “2” nor “5”, then it shall be set to “5”.

13.11 *User-Initiated State Change Requests* — The user may request an ARAMS state change at any time. The host requests a state change through the ARAMS message ARAMSStateChange (Section 15.5), specifying an ARAMS Substate Code or the special manufacturing code “0000”. The operator uses the human interface to request a state change, and this shall result in the specification of a valid ARAMS Substate Code (Section 9.4) or of the manufacturing code.

4.1.11 The equipment shall deny the request if the specified code is not valid or if the user specifies a code for manufacturing when any exception condition exists that would prevent the equipment from performing its intended function.* Otherwise, Transition 10 occurs, regardless of whether the new ARAMS Substate Code is the same or different from the value in ARAMSState representing the current state.

* NOTE: See SEMI E10 definitions.

4.1.12 If the user specifies a manufacturing state, and the equipment accepts the state change request, then Transition 10 occurs and is immediately followed by Transition 2. Transition 2 always occurs in conjunction with Transition 10 and does not generate a report separately from Transition 10. It is regarded as an extension of Transition 10 where the equipment determines the specific state/substate based on its internal status at the time.

4.1.13 At Transition 2, the equipment determines the new state as either PRODUCTIVE or STANDBY, based on its internal condition at the time. It is prohibited from transitioning to PRODUCTIVE unless its productive criteria have been met and it is busy performing its intended function. The current value of ARAMSState is moved to PrevARAMSState, and the ARAMS Substate Code for the new state is placed in ARAMSState. The event report associated with Transition 10 shall be generated after the equipment has determined the new state at Transition 2 and updated the appropriate variables.

4.1.14 The user may optionally specify a symptom identifier and text, which are saved in the variables SymptomID and SymptomText (Section 11.2). If not provided, SymptomID is set to zero and SymptomText is set to a zero-length text string. The equipment does not otherwise set these values.

4.1.15 The user also may optionally enter comments that are stored in DowntimeData when selecting a transition to UNSCHEDULED DOWNTIME from a manufacturing state. Otherwise, DowntimeData is set to a zero-length text string.

13.12 *Production and Standby Substates* — Transitions 2, 3, and 4 are made automatically by the equipment. User requests to change to PRODUCTIVE or STANDBY are accepted as methods of setting the current substate of PRODUCTIVE or STANDBY. However, within the MANUFACTURING superstate, the equipment is responsible for determining when its production criteria are satisfied.

4.1.16 Unless the equipment has information as specified in this section concerning the appropriate substate of PRODUCTIVE or STANDBY, then the equipment shall select the appropriate default ARAMS Substate Code ("1000" or "2000", defined in Section 9.2).

4.1.17 To provide substate refinements important to the user, the equipment shall remember the last ARAMS Substate Code specified by the user for the PRODUCTIVE state at Transition 10 and use this as a default PRODUCTIVE substate value PrdState (Section 11.2) until a new value is specified. This value shall be used for the new (current) ARAMS state/substate information in the variable ARAMSState whenever the equipment transitions to PRODUCTIVE.

4.1.18 The equipment normally uses the default code for all transitions to STANDBY ("2000"). However, when the user specifies an ARAMS Substate Code for STANDBY with a substate, this value is to be used at the next following transition to STANDBY at Transition 2 (if the equipment then transitions to STANDBY) or at the subsequent occurrence of Transition 4 (if the equipment transitioned to PRODUCTIVE at Transition 2).

4.1.19 This value is discarded (or reset to "2000") after a single use or upon any other transition that intervenes between the Transition 10 where it is specified and the time it is applied. This includes intervening occurrences of a new Transition 10 as well as of Transitions 11, 1, and others.

4.1.20 It is not important that the equipment provide the user access to this value except when it is stored in ARAMSState.

13.13 *Equipment-Selected Substates* — Equipment may provide an optional capability to select substates of PRODUCTIVE and STANDBY. In this case, the equipment shall provide a user-configurable variable SubstateSelect that enables and disables this capability (Section 11.4.7).

4.1.21 Substate selection by the equipment is subject to the requirements specified in Section 16.5. If the user has selected a substate of PRODUCTIVE or STANDBY, the equipment is restricted to that substate or extensions of that substate. Note that a user-selected substate of STANDBY is applied at most once.

4.1.22 The capability of selecting substates includes the ability to select a new substate of the current state. Transistions 12 and 13 shall be used for this purpose.

4.1.23 When transitioning to STANDBY, normally at the completion of a process cycle, multiple standby conditions may exist. For example, the equipment may need both material and instructions from the host. In general, a substate code of 2200 (SBY/No product) shall take precedence over 2100 (SBY/No operator) and 2500 (SBY/No host).

4.1.24 If the standby condition represented by the currently selected substate clears, then the equipment shall either select a new substate of STANDBY (Transition 12) or transition from STANDBY (Transition 3).

4.1.25 Additional rules for determining substates may be specific to a type of equipment and may be specified by standards defining the capabilities for that type of equipment.

4.1.26 The equipment supplier shall document the prioritization of standby conditions used by the equipment and of the basis for selecting substates of PRODUCTIVE.

13.14 *Equipment-Detected Exceptions* — The equipment may detect exceptions in any ARAMS state. However, only those exceptions that occur during an uptime state are of interest to the host.

4.1.27 Transitions 5 and 7 indicate the equipment detects a fault condition and transitions to UNSCHEDULED DOWNTIME from STANDBY and PRODUCTIVE respectively. Transitions 6 and 8 are provided to allow the equipment to return to its prior state and are discussed in the following section.

4.1.28 When the equipment detects an exception condition during PRODUCTIVE or STANDBY, so that it is unable to perform its intended function, it shall immediately transition to UNSCHEDULED DOWN-TIME. The identifier of the associated alarm is stored in DowntimeAlarm, and any description text associated with that alarm is stored in DowntimeAlarmText. The equipment supplier may provide additional descriptive text that would be useful in diagnostics or analysis in DowntimeData. In the case of the failure of a component, attributes of that component (e.g., serial number, installation date, lifetime cycles) is desirable.

4.1.29 If the equipment detects an exception condition in non-uptime states, it shall not initiate a state change. Exceptions occur for many reasons in non-manufacturing states. For example, the equipment may be improperly or incompletely installed, or it may be deliberately pushed past its limits.

4.1.30 *Fault Detection in ENGINEERING* — The equipment may also provide optional capability of transitioning to UNSCHEDULED DOWNTIME from ENGINEERING (Transition 12) when it detects an exception. Alarm-related variables in this case are handled in the same way as described above.

4.1.31 The user shall be able to enable and disable this capability with the user-configurable variable EngInterrupt (Section 11.4.1).

13.15 *Equipment-Initiated Recovery* — Equipment-initiated transitions 6, 8, and 13 from UNSCHEDULED DOWNTIME are provided to allow equipment to recover from an equipment-detected fault when the operator intervenes, corrects the fault, and indicates the process can be recovered. For Transition 6 to occur, the equipment shall be able to resume processing without degradation of the process or the material. The equipment is responsible for ensuring the safety of persons, material, and for the equipment itself.

4.1.32 Transition 13 is required if Transition 12 is supported and is prohibited otherwise.

13.15.1 *Automatic Recovery* — Equipment may also provide optional capabilities to recover automatically from transient faults that clear spontaneously. Automatic recovery at Transitions 6, 8, and 13 shall be separately enabled and disabled using the user-configurable variables PrdRecovery (Transition 6), SbyRecovery (Transition 8), and EngRecovery (Transition 13). Equipment is otherwise prohibited from using Transitions 6, 8, or 13 to recover without explicit operator approval.

4.1.32.1 Automatic recovery to manufacturing and automatic recovery to ENGINEERING are regarded as two separate capabilities.

4.1.32.2 If EngInterrupt (Section 11.4.1) is not supported or is disabled, then Transition 13 is prohibited.

14 Requirements for Compliance

This section summarizes the requirements for compliance to ARAMS that are defined in this document.

14.1 *Fundamental Requirements* — Compliance to ARAMS requires certain capabilities that are defined by other standards.

14.1.1 *Event Notification* — A standard method for notifying the host that an event of interest has occurred and for providing specific information related to the event.

14.1.2 *Clock Services* — Provision of a real-time date/time clock with methods for setting and reading the clock from the host.

14.1.3 *Read-Only Data Access* — A standard method for the host to obtain the current values of selected status variables and constants specified in Sections 11.2 and 11.3.

14.1.4 *User-Configurable Data Access* — A standard method for the host to change the values of selected variables defined in Sections 11.4 and 11.5.

14.1.5 *Alarm/Exception Management* — A standard method for notifying the host of abnormal events and/or conditions.

4.1.32.3 In addition to the requirements defined in other standards, the following, defined by ARAMS, are required for compliance to ARAMS:

14.1.6 *ARAMS State Model* — Conformance to the behavior of the ARAMS state model defined in Section 8.3.

14.1.7 *ARAMS State Transition Notification* — Using standard event report mechanisms (above), the host shall be notified of all state transitions as described in Sections 12 and 16.1.

14.1.8 *ARAMS Substate Codes* — Support for ARAMS Substate Code formats, used for equipment variables and service parameters defined in Section 9.

14.1.9 *ARAMS Status Data* — Support for all status variables defined in Section 11.2.

14.1.10 *ARAMS Constant Data* — Support for all data constants and for the user-configurable variable EqpName defined in Sections 11.3 and 11.4.

14.1.11 *ARAMS Event Report Data* — Support for the requirements of Section 12 requires either the provision of pre-defined event reports or provision of the Dynamic Event Report Configuration capability that allows the host to dynamically modify the equipment event reporting setup and define the content of reports for each event.

14.1.12 *Host State Change Request* — Support for the ARAMSStateChange service, defined in Sections 15.1, 15.2, and 15.5.

14.1.13 *Estimation of Powerdown Time* — Provision of a method for maintaining PowerdownTime as an estimate of the time of powerdown to an accuracy within ± one minute.

14.1.14 *ARAMS Behavioral Requirements* — Conformance to all requirements in Section 16 except those identified as optional or requiring optional variables.