Table 1 Object State Definitions and Query Table Example

State	*Definition*	*Query for State Via*
ENGINE RUNNING	In this state, the automobile's engine is running.	boolean isEngineRunning ( ); sent to instance of Automobile. Returns true.

Table 2 Object State Transition Table Example

Current State	*Trigger*	*New State*	*Action*
ENGINE STOPPED	turn ignition key to right	ENGINE RUNNING	tachometer indicates positive RPM
ENGINE RUNNING	turn ignition key to left	ENGINE STOPPED	tachometer indicates zero RPM
LIGHTS OFF	turn light switch on	LIGHTS ON	driving at night is enabled
LIGHTS ON	turn light switch off	LIGHTS OFF	driving at night disabled

7.4.4 *Textual Specification Language*

7.4.4.1 The textual notations used to specify the framework in this specification, include the following:

- Interface Definition Language (IDL).
- Interface Definition Format.

7.4.4.2 *Interface Definition Language (IDL)*

7.4.4.2.1 Specifications of CIM Framework components are composed mainly of interfaces. The specifics of each interface, while also represented in the UML-based Component Information Model, are rigorously specified using OMG IDL. The IDL for CIM Framework interfaces shall be complete and consistent as verified by automated IDL compilers. The IDL portion of SEMI specifications will be available in a text file format to facilitate such validation and use with IDL compiler technologies.

7.4.4.2.2 This explanation of IDL is addressed to its use as a rigorous specification tool. The considerations of mapping the specifications to an implementation infrastructure are addressed in SEMI E96.

7.4.4.2.3 *Common Object Broker Architecture (CORBA)* [CORBA][CIMArch] defines the architecture which enables and regulates interoperability between objects and applications across heterogeneous languages and computer boundaries. In all Object Management Group (OMG) specifications, services are defined as object interfaces expressed in the OMG's *IDL*. *CORBA* standards define *IDL* and its mapping to implementation languages (for example, *C*, *C++*, *Smalltalk*, and *Java*).

7.4.4.2.4 *IDL* is a compilable language that describes the operations that are specified for an interface. The notation is independent of the language in which the methods that implement an interface's operations are written. This goal is achieved by mapping between the *IDL* syntax and whatever language is used to implement client and server objects. Because *IDL* is designed purely for interface specification, it omits the flow control and operator constructs of an implementation language. Object classes can implement an interface differently as long as their behavior conforms to the interface specification.

7.4.4.2.5 *IDL* obeys the same lexical rules as *C*, while introducing a number of keywords specific to a distributed system. As *IDL* is mapped to object-oriented languages, new constructs will appear. A brief discourse on some of the keywords and concepts used in this specification follows. In the examples, words in italics are user supplied, others are *IDL*-defined keywords.

7.4.4.3 *Interface Specification Format*

7.4.4.3.1 The OMG defines an object's interface as "a listing of the operations and attributes that an object provides. This includes the signatures of the operations, and the types of the attributes. An interface specification ideally includes the semantics as well" [OMA]. The CIM Framework Specification builds upon this definition to provide additional semantic information for an interface. These semantics are captured in an interface description. This section describes the format of an interface. Each description of an interface follows this format. The format includes the following:

- Name — The capitalized noun following the word "Interface:"
- Inherited Interface — The capitalized noun after the words "Inherited Interface:"
- Description — A definition of the interface giving its form and function.

- Exceptions — An IDL specification for reporting user-defined, framework-related error conditions.
- Published Events — The name of the event structure that must be placed on an event channel. The event structure identifies the event through a subject field. The subject is composed of the component and interface issuing the event and data describing the event, and filterable and non-filterable information. Events are defined at the interface level; neither posting services nor subscribers are identified. Thus, events are not tied to specific services and may be the result of an internal (to the component) computation.
- Provided Services — A list of publicly available services provided by this interface. In other words, a list of non-private, named operations. Each is given by a description in comment form (i.e., /*....*/) followed by its representation in the IDL syntax.
- Contracted Services — A table of framework services provided by other interfaces that are being used by public and/or private services in this interface. These methods must be available in order for the documented interface to provide its described services. Changes to contracted services may result in changes to the behavior of the interface making use of these services.
- State Model — See Section 7.4.3.5 for details.

7.4.4.3.2 If no Provided Services are defined for a particular category, then "No public interfaces" will appear after the category identifier. If no Exceptions, Published Events, Contracted Services, or Dynamic Model are provided, then the word "None" will appear. State Transition Tables occur only in conjunction with Dynamic Models.

7.4.4.4 *Interface Specification Example*

7.4.4.4.1 Table 3 provides a complete illustration of *IDL* and CIM Framework interface specification format usage within this specification. Words in italics are user-supplied, others are either *IDL* or CIM Framework-defined keywords.

8 Conformance to CIM Framework Domain Specifications

8.1 The objective of the CIM Framework is to speed the creation, use and improvement of a manufacturing execution system for a semiconductor wafer fabrication factory by enabling the integration of disparate components into a cohesive system. The CIM Framework achieves this objective by specifying a domain model for MES components. MES component suppliers use the specification to help establish the boundaries and interfaces of their components. Component customers first use the specification to assess the capabilities of individual components and then to facilitate the integration of components into a working system.

8.2 The CIM Framework increases the value of the MES components by enhancing their qualities of interoperability, substitutability and extensibility.

- Interoperability is the ability of components to work together through compatible interfaces.
- Substitutability implies the option to swap one component with another because they support the same interfaces.
- Extensibility means the planned capability to add functionality to a component, again by leveraging the support for predefined interface specifications.

8.3 Given both the scope and objectives of the CIM Framework, conformance to the Framework can not be reduced to a simple "yes or no" proposition. Rather, component customers must assess component conformance on a case-by-case basis. Potential buyers assess conformance in terms of how well a component supplier demonstrates use of the CIM Framework specification to enable rapid component integration.

8.4 With this background, here then are factors for consumers to consider in assessing MES components and for suppliers to comprehend when building components. These factors form the basis for communication between buyers and sellers of components regarding conformance to the CIM Framework specifications.

Table 3 Interface Specification Example

/* Comments are set between slashes and asterisks */

Interface: *FrameworkObject2*

Inherited Interface: *FrameworkObject1*

Description: The example defines the interface for *FrameworkObject2*, which inherits from *FrameworkObject1*.

Exceptions:

/* The following portrays the syntax used to describe exceptions for this interface. *ObjectType* and *instanceName* specify an object instance (supplementary information) returned with the exception. */

```
exception ExceptionName { ObjectType instanceName };
```

Published Events: *NamedEvent*

Provided Services:

/* The following defines the read/write methods for AttributeName1. */

```
ObjectType getAttributeName1 ( );
void setAttributeName1 (in ObjectType parameterName);
```

/* The following defines a method for readonly *AttributeName2* */

```
ObjectType getAttributeName2 ( );
```

/* The following says operationName1 is a local operation returning an object of the class ObjectTypeReturned. */

```
ObjectTypeReturned operationName1 ( );
```

/* The following says *operationName2* is a local operation returning an object of the class *ObjectTypeReturned* with an argument *instanceName* of the object type *ObjectType*. In addition, there is an operation-specific exception, *E*, that may be raised by this operation. */

```
ObjectTypeReturned operationName2 (in ObjectType instanceName)
        raises (E);
```

The type definitions follow the following format:

/* Type Declarations: */

/* The following specifies (types) the *ObjectType* for the named *datatype*. */

```
typedef ObjectType datatype ;
```

/* The following specifies a sequence (collection) of *ObjectType* for *ObjectTypeSequence*. */

typedef sequence<*ObjectType*> *ObjectTypeSequence* ;

- **CIM Framework component packaging:** Component suppliers must explain how their component is packaged relative to the corresponding CIM Framework component(s). Note, however, that suppliers may choose to provide multiple CIM Framework components as an integrated package. In this case the complete package can be assessed relative to the combination of services provided by the combined set of CIM Framework components. Obviously, a consumer must also assess the benefits of the integrated component relative to the reduction in the ability to substitute components within the integrated package.
- **CIM Framework objects and interfaces:** Component suppliers must describe how their objects and methods support the CIM Framework component interfaces. This includes describing the object methods available in comparison to the interfaces specified in the CIM Framework. Note that this interface specification question encompasses specific operations, the operations' arguments, the exceptions returned, and the events published.
- **Object behavior:** Component suppliers must document object behavior so component consumers can understand the purpose and consequences of specific methods. The CIM Framework specifies behavioral semantics for components using a variety of representations such as state models, information models showing relationships, interaction models, and text-based

comments. Component suppliers need to explain how their objects conform to the CIM Framework behavioral semantics.

8.5 As these factors indicate, assessing the CIM Framework conformance of supplied components takes more than just verifying the existence of specific objects and methods. The key issue is whether a component supplier provides both the software and its associated conformance information needed to enable the use and integration of the component.

9 Related Documents

9.1 The following documents describe programs, standards, and guidelines used in the development of the CIM Framework specification.

9.1.1 *SEMATECH Documents*

Advanced Process Control Framework Initiative (APCFI) 1.0,6/27/97 (SEMATECH - Technology Transfer #97063300A-ENG):CIM Framework Enhanced Machine Component Communications Driver (MCCD) Final Report (SEMATECH - Technology Transfer #97073323A-TR).

C++ Reference Implementation for the Computer Integrated Manufacturing (CIM) Application Framework: Release 2, 1/4/96 (SEMATECH - Technology Transfer #95082944B-ENG).

Computer Integrated Manufacturing (CIM) Application Framework Validation Project: Lessons Learned During the Automation Software Systems Project (SEMATECH - Technology Transfer #94102568A-ENG).

Computer Integrated Manufacturing (CIM) Development Manual 1.1 - Volumes 1 and 2 (SEMATECH - Technology Transfer #91120794B-ENG).

Computer Integrated Manufacturing (CIM) Framework Member Validation Project (FMVP): Phase II Final Report (SEMATECH - Technology Transfer #96013061A-TR).

Evolution of WorkStream for Preventive, Predictive Maintenance (PM) at SEMATECH (SEMATECH - Technology Transfer #95092966A-TR).

Real-Time Dispatcher (RTD) Computer Integrated Manufacturing (CIM) Framework Conformance and Integration Studies (SEMATECH - Technology Transfer #96023088A-ENG).

Results of the AutoSimulations and TI/WORKS Integration Feasibility Study (SEMATECH - Technology Transfer #95092981A-ENG).

SEMATECH Workbench for Integrated Modeling (SWIM) Enhanced Prototype Functional Specification 5.0, 12/2/93 (SEMATECH - Technology Transfer #93112072A-ENG).

Semiconductor Generic Manufacturing Model. (SEMATECH - Technology Transfer #91090704A-ENG).

Semiconductor Generic Manufacturing Requirements Specification (SEMATECH - Technology Transfer #91090703A-ENG).

Strategic Cell Controller (SCC) Program Repository Contents Guide 1.1 SEMATECH Factory Integration Technologies (FIT) Project (SEMATECH - Technology Transfer #93091827B-XFR).

Strategic Computer Integrated Manufacturing (CIM) Computing Environment Specifications. (SEMATECH - Technology Transfer #92010916A-ENG)

Technical Summary of CIM Framework-Based Integration of ASI Real-Time Dispatcher and IBM Legacy Systems (SEMATECH - Technology Transfer #96093180A-TR).

SEMI E82-0705
SPECIFICATION FOR INTERBAY/INTRABAY AMHS SEM (IBSEM)

This specification was technically approved by the global Information & Control Committee. This edition was approved for publication by the global Audits and Reviews Subcommittee on April 7, 2005. It was available at www.semi.org June 2005 and on CD-ROM in July 2005. Originally published September 1999; previously published November 2004.

1 Purpose

1.1 This standard establishes a Specific Equipment Model (SEM) for interbay and intrabay AMHS transport equipment (IBSEM). The model consists of equipment characteristics and behaviors that are to be implemented in addition to the SEMI E30 fundamental requirements and selected additional capabilities. The intent of this standard is to facilitate the integration of IBSEM equipment into an automated (e.g., semiconductor fabrication and flat panel display) factory. This document accomplishes this by defining an operational model for IBSEM equipment as viewed by a factory automation controller (Host). This definition provides a standard host interface and equipment operational behavior (e.g., control, state models, data reports, and reporting levels). Several topics require additional activity that are within the scope of this standard: traffic management characteristics (queuing), parallel interface for carrier transfer (SEMI E23), transport system controller architecture, and delivery of the transfer unit.

2 Scope

2.1 The scope of this standard is limited to the usage and description of interbay and intrabay AMHS transport equipment (OHT, OHS, RGT, AGT, DWC) as perceived by a SEMI Equipment Communications Standard 2 (SECS-II) host that complies with the GEM model (as specified in §13). It defines the view of the equipment through the SECS communication link. It does not define the internal operation of the equipment. It includes a specific transfer command state model and transport system controller state model as the basis for all equipment of this class.

2.2 This document assumes that the GEM fundamental requirements and selected additional capabilities (as specified in §13) have been implemented on the IBSEM equipment. It expands the GEM standard requirements and capabilities in the areas of state models (TSC, transfer command, vehicle and carrier state models), collection events, alarm documentation, remote commands, data item variables, and material movement.

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 *Evaluation of SEMI E32 (MMM)*

3.1.1 The concepts defined in SEMI E32 were analyzed and included where applicable to the IBSEM, but the GEM model was used as the basis for IBSEM requirements definition.

3.2 *Interbay and Intrabay AMHS Transport Equipment Types*

3.2.1 This standard is targeted at the different types of 300 mm and interbay and intrabay AMHS transport equipment. The term *IBSEM equipment* refers to all types of transport equipment. The equipment types have fundamental mechanical differences:

3.2.1.1 *Overhead Hoist Transport (OHT)* — An overhead rail guided transport system positioned for vertical access to SEMI E15.1 compliant ports.

3.2.1.2 *Over Head Shuttle (OHS)* — An overhead rail guided transport system (monorail) positioned for access to stocker automated interbay input and output ports. The OHS vehicle may or may not contain a transfer agent.

3.2.1.3 *Rail Guided Transport (RGT)* — A ground-based rail guided transport system positioned for access to SEMI E15.1 compliant ports.

3.2.1.4 *Automated Guided Transport (AGT)* — A ground-based transport system with automated guidance (i.e., no rail guidance). Automated guidance system allows vehicles to access SEMI E15.1 compliant ports.

3.2.1.5 *Direct WIP Conveyer (DWC)* — An overhead transport system, based on direct WIP roller conveyers. No vehicles are used for point to point delivery. The conveyers are positioned for vertical access to SEMI E15.1 ports.

3.2.2 Transport vehicles may contain zero or more internal buffers for carrier transport. If mechanically feasible, the transport system may acquire or deposit carriers simultaneously. If transported in a safe manner, carrier transport may occur while occupying the acquire/deposit transfer port(s) of the transport vehicle (e.g., a single position hoist vehicle). In the context of this standard, a "vehicle" on a DWC is defined as a single carrier in the transport system.

3.3 *Physical Layout Limitations*

3.3.1 The equipment controlled by a single TSC must allow for a carrier to be transported from any given source port to any destination port via a single transfer command without the assistance of an external device (manual or automated). In other words, if a source port and a destination port are controlled by a TSC, there must not exist a physical or logical barrier that prevents a carrier from being moved between the two ports. This assumes that the type of carrier (FOUP, Reticle Pod, etc.) is permitted at the source and destination ports.

4 Referenced Standards

4.1 *SEMI Standards*

SEMI E4 — SEMI Equipment Communications Standard 1 Message Transfer (SECS-I)

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

SEMI E23 — Specification for Cassette Transfer Parallel I/O Interface

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

SEMI E32 — Material Movement Management (MMM)

SEMI E37 — High-Speed SECS Message Services (HSMS) Generic Services

SEMI E84 — Specification for Enhanced Carrier Handoff Parallel I/O Interface

4.2 *Other References*

Harel, D., "Statecharts: A Visual Formalism for Complex Systems," Science of Computer Programming 8 (1987) 231-274.[1]

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

5 Terminology

5.1 *Abbreviations and Acronyms*

5.1.1 *AGT* — Automated Guided Transport

5.1.2 *AMHS* — Automated Material Handling System

5.1.3 *DWC* — Direct WIP Conveyor

5.1.4 *FOUP* — Front Opening Unified Pod

5.1.5 *GEM* — Generic Equipment Model

5.1.6 *ITS* — Interbay or Intrabay Transport System

5.1.7 *OHS* — Over Head Shuttle

5.1.8 *OHT* — Overhead Hoist Transport

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

5.1.9 *PGV* — Person Guided Vehicle

5.1.10 *RGT* — Rail Guided Transport

5.1.11 *TCP/IP* — Transmission Communication Protocol/ Internet Protocol

5.1.12 *TSC* — Transport System Controller

5.2 *Definitions*

5.2.1 *active vehicle* — a vehicle in the transport system that contains a robot or other transfer agent for providing the acquiring (loading) and depositing (unloading) actions.

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

5.2.3 *carrier* — a container with one or more fixed positions for holding substrates. Examples of carriers include FOUPs and open cassettes.

5.2.4 *FOUP* — a closed carrier for holding wafers.

5.2.5 *host* — the factory computer system, or an intermediate system, that represents the factory and the user to the equipment. Refers to the system that controls or supervises the Transport System Controller (TSC) throughout this document.

5.2.6 *internal buffer* — locations within the equipment to store carriers. These locations exclude load ports.

5.2.7 *internal transfer port* — a specific type of *transfer port,* which is internal to a single TSC domain. As an example, this location may be used to transfer carriers among different vehicles in a single TSC domain.

5.2.8 *load port* — the interface location on the equipment where carriers are delivered.

5.2.9 *open cassette* — an open structure that holds one or more wafers.

5.2.10 *passive vehicle* — a vehicle in the transport system that does not contain a robot or other transfer agent for providing the acquiring (loading) and depositing (unloading) actions. The vehicle simply contains a position(s) to carry the transfer unit. The loading and unloading action must be accomplished at the load or unload port by a different system (e.g., stocker port robot).

5.2.11 *process equipment* — equipment used to make semiconductor devices. This excludes metrology and material handling equipment.

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

5.2.13 *transfer port* — point on the transport system at which a change of equipment ownership of the carrier occurs. See also *internal transfer port.*

5.2.14 *transfer unit* — the element of movement (assemblage of carriers) of the ITS that consists of a maximum number of carriers allowed in a specific transfer command:

- AA is the maximum number of carriers allowed for acquire at the transfer source.
- BB is the maximum number of carriers allowed for deposit at the transfer destination.
- CC is the maximum number of carriers allowed for transfer in one transport vehicle.
- The maximum size of the transfer unit is the minimum of AA, BB, and CC.

5.2.15 *Transport System* — a transport system dedicated to one or more bays in the factory and responsible for transferring carriers to production equipment, from production equipment, from production equipment to production equipment or from stocker to stocker. TS consists of the physical units of the system (e.g., vehicles, nodes, docking stations), the low-level unit controllers, and a system-level controller. TS excludes factory floor storage systems (stockers), but includes any short-term storage integral to the system, such as storage locations within an overhead track system that are accessible only to units of the particular TS.

5.2.16 *Transport System Controller* — interbay or intrabay Transport System Controller that communicates with the Factory Host and represents the system as the equipment.

5.2.17 *Transport System Equipment* — an individual transport system viewed as a single piece of equipment, with distributed components and distributed control. The TS controller communicates with the host using HSMS and GEM and represents the system as an equipment. The factory may require more than one type of transport system.

5.2.18 *transport unit* — a physical component of a transport system, such as a vehicle, node, or docking unit.

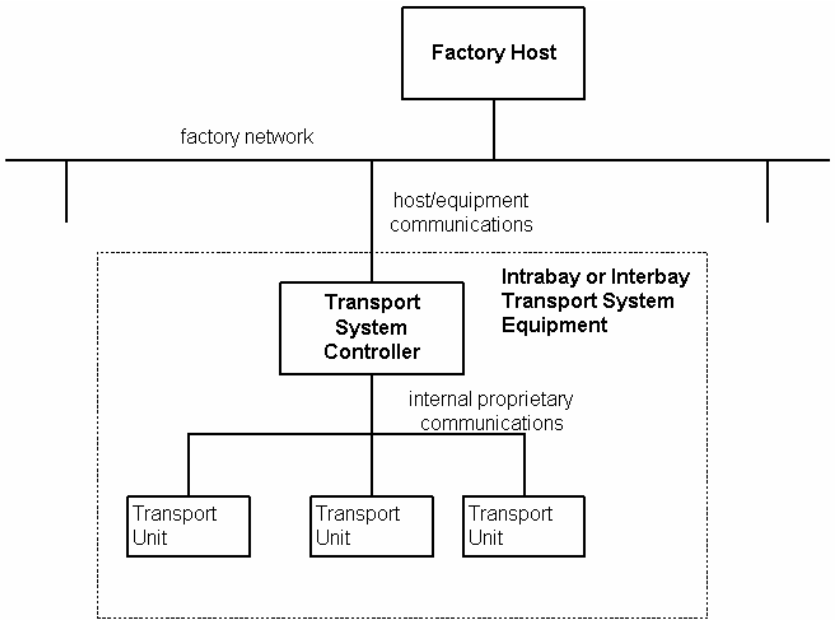

Figure 1
Example of Transport System Equipment

6 Communication Requirements

6.1 It is required that any IBSEM compliant equipment follow the Communications State Model in SEMI E30. In addition, IBSEM compliant equipment shall support either the High-speed SECS Message Services Single-Session Mode (SEMI E37 and SEMI E37.1, HSMS and HSMS-SS) communication standard or SEMI Equipment Communications Standard 1 Message Transfer (SEMI E4, SECS-I) communication standard.

7 State Models

7.1 *State Model Requirements*

7.1.1 The state models included in this standard are a requirement for IBSEM equipment. This standard requires implementation of all SEMI E30 state models (such as control, communication, on-line/off-line, etc. according to the GEM capabilities required per §13). A state model consists of a state model diagram, state definitions, and a state transition table. All state transitions in this standard, unless otherwise specified, shall correspond to collection events.

7.1.2 A state model is the host's view of the equipment, and does not necessarily describe the internal equipment operation. All IBSEM state model transitions shall be mapped into the appropriate internal equipment events that satisfy the requirements of those transitions. In certain implementations, the equipment may enter a state and have already satisfied all of the conditions required by the IBSEM state model for transition to another state. The equipment makes the required transition without any additional actions in this situation.

7.1.3 Some equipment may need to include additional substates other than those in this standard. Additional substates may be added, but shall not change the IBSEM defined state transitions. All expected transitions between IBSEM states shall occur.

7.2 *TSC State Model*

7.2.1 *TSC State Model Requirements*

7.2.1.1 The purpose of the Transport System state model is to provide information to the host regarding the overall status of the Transport System. The TSC state model is valid when the SEMI E30 (GEM) state is ON-LINE. The TSC state model is **not** valid when the SEMI E30 (GEM) state is OFF-LINE. Since a transport system may consist of many components (e.g., vehicle, robot arm, ID reader, etc.), it may be possible to continue ON-LINE operation when the operation mode of some transport components (as viewed by the TSC) is a manual state. The details of what happens when individual components of the transport system enter a manual state are specific to the IBSEM equipment supplier. When the SEMI E30 Control state changes from OFF-LINE to ON-LINE, the TSC State Model is started from the TSC INIT state.

7.2.2 *TSC State Model*

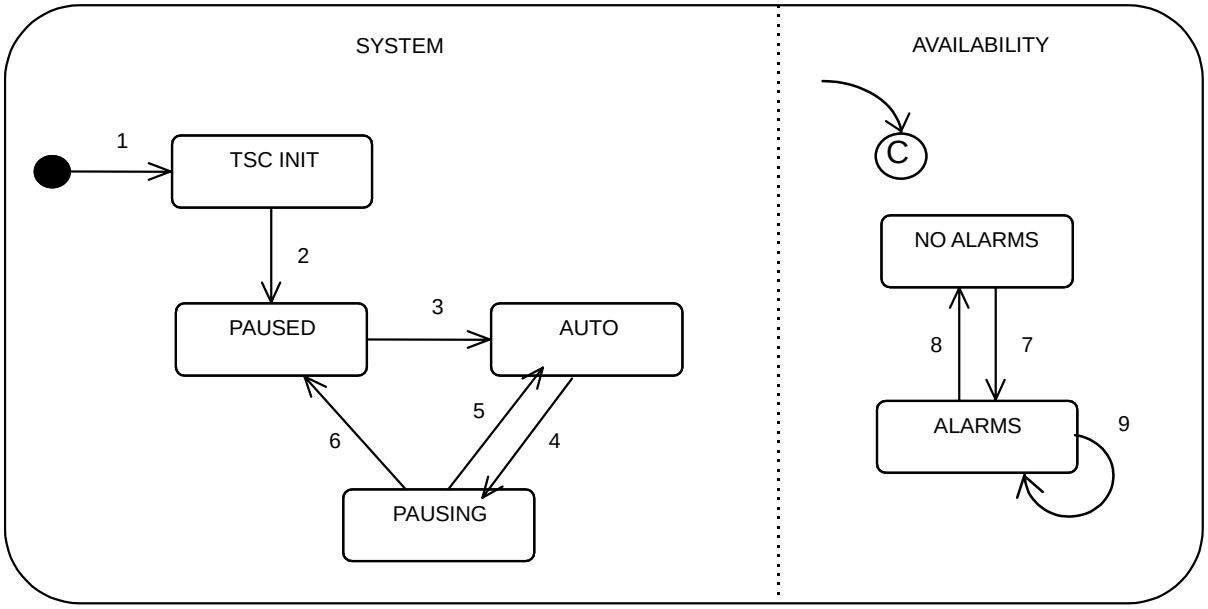

Figure 2
Generic IBSEM TSC State Model Diagram

7.2.3 *TSC State Definitions*

7.2.3.1 *TSC INIT* — TSC initialization of TS components is occurring. This is a non-operational state. No commands from the host will be processed or queued. The system will not move out of this state if there are vehicles actively loading or unloading carriers at ports. These vehicles must be manually or automatically recovered before moving on to the next state.

7.2.3.2 *PAUSING* — A system PAUSE command has been received and is being processed. All vehicles that are currently loading or unloading will continue until the load/unload is complete. Vehicles that are currently moving may continue to move but they must not begin a load or unload. TRANSFER commands are accepted and queued. All status requests will be processed. The RESUME command will also be processed.

7.2.3.3 *PAUSED* — No vehicles are in the process of loading or unloading a carrier at a port, but vehicles may still be moving. TRANSFER commands are accepted and queued. All status requests will be processed. The RESUME command will also be processed.

7.2.3.4 *AUTO* — System is in the normal operational state. Commands are actively processed.

7.2.3.5 *NO ALARMS* — There are no alarms present in the system.

7.2.3.6 *ALARMS* — There are one or more alarms present in the system, but the TSC is still capable of normal processing since several components may remain unaffected by the alarm situation.

7.2.4 *TSC State Transition Table*

Table 1 TSC State Transition Table

Transition #	*Previous State*	*Trigger*	*New State*	*Actions*	*Comments*
1	None	TSC Initiation.	TSC INIT	S6F11 TSCAutoInitiated	System runs through its startup sequence.
2	TSC INIT	System started up successfully. All loads and unloads are complete.	PAUSED	S6F11 TSCPaused	System ready.
3	PAUSED	TSC is resumed.	AUTO	S6F11 TSCAutoCompleted	System will now perform all commands. Stopped Vehicles will resume normal motion.
4	AUTO	TSC is requested to pause.	PAUSING	S6F11 TSCPauseInitiated	Vehicles that are stopped stay stopped. Vehicles that are moving stop at the next logical point without proceeding.
5	PAUSING	TSC is resumed.	AUTO	S6F11 TSCAutoCompleted	System will now perform all commands. Stopped Vehicles will resume normal motion.
6	PAUSING	All carrier loads and unloads are completed. No new acquires or deposits will occur. Outstanding acquires and deposits will complete.	PAUSED	S6F11 TSCPauseCompleted	System will accept and queue new commands but will not execute them.
7	NO ALARMS	Alarm Set.	ALARMS	S6F11 AlarmSet	System can process normally for transport components that are unaffected by the alarm.
8	ALARMS	Last remaining alarm cleared.	NO ALARMS	S6F11 AlarmCleared	
9	ALARMS	Alarm Set.	ALARMS	S6F11 AlarmSet	Alarm occurs when there is already an outstanding alarm.

7.3 *TRANSFER Command State Model*

7.3.1 *TRANSFER Command State Model Requirements*

7.3.1.1 The TRANSFER command state model serves as the SEMI E30 Processing State Model. The purpose of the TRANSFER command state model is to provide information to the host regarding the control of the TRANSFER command. The TRANSFER command allows the host to manage interbay or intrabay delivery and scheduling. The control of each TRANSFER command must independently support the TRANSFER command state model.

7.3.2 *TRANSFER Command State Model Diagram*

7.3.2.1 The TRANSFER command state model is detailed for IBSEM equipment in Figure 3.

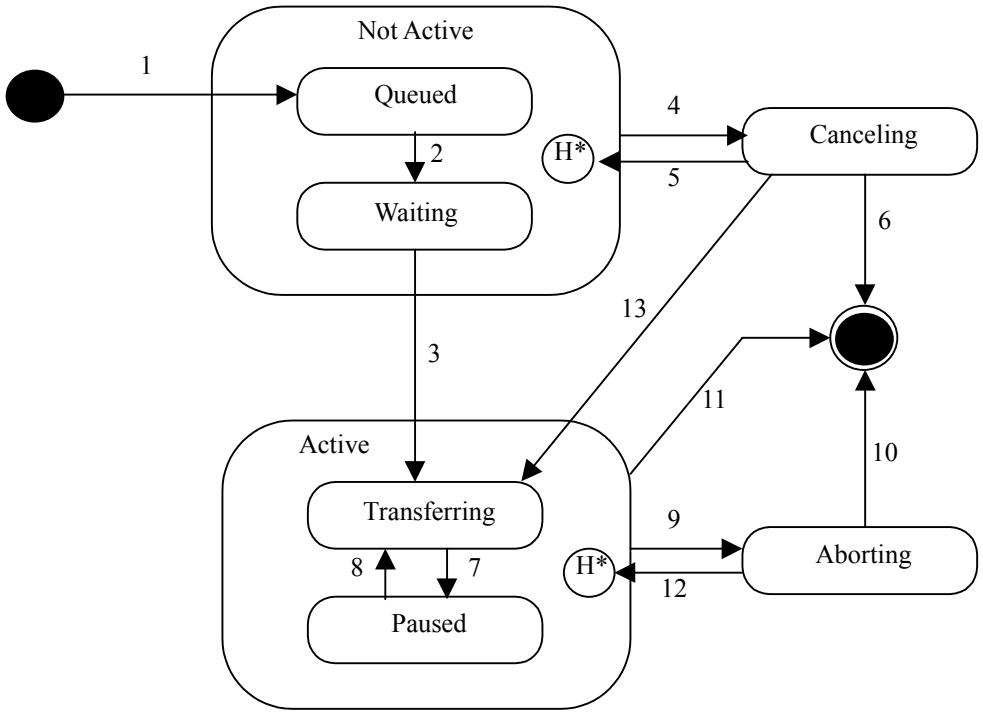

Figure 3
Generic IBSEM TRANSFER Command State Model Diagram

7.3.3 *TRANSFER Command State Definitions*

7.3.3.1 *NOT ACTIVE* — The transfer unit is not involved in the physical aspect of the TRANSFER command. It is denoted by the time spanned by the queuing of the TRANSFER command to the moment just prior to the acquire of the first carrier in the transfer unit.

7.3.3.2 *QUEUED (NOT ACTIVE sub-state)* — TSC has acknowledged and queued the TRANSFER command. TRANSFER command has not been initiated.

7.3.3.3 *WAITING (NOT ACTIVE sub-state)* — TRANSFER command has been initiated. A vehicle is on its way to the source location to acquire the transfer unit.

7.3.3.4 *ACTIVE* — The transfer unit is involved in the physical aspect of the TRANSFER command. It is denoted by the time spanned by the acquire of the first carrier in the transfer unit to the deposit of the last carrier in the transfer unit.

7.3.3.5 *TRANSFERRING (ACTIVE sub-state)* — The transfer command is actively being executed by the transport equipment.

7.3.3.6 *PAUSED (ACTIVE sub-state)* — The transfer command is not actively being executed by the transport equipment.

7.3.3.7 *CANCELING* — The TRANSFER command cancel procedure is being performed to terminate a transfer command which never entered the ACTIVE state (either QUEUED or WAITING). This state is entered via a CANCEL remote command.

7.3.3.8 *ABORTING* — The TRANSFER command abort procedure is being performed to terminate a transfer command which has entered the ACTIVE state. This state can only be entered via an ABORT remote command.

7.3.4 *TRANSFER Command State Transition Table*

Table 2 Transfer Command State Transition Table

Transition #	*Previous State*	*Trigger*	*New State*	*Actions*	*Comments*
1	None	The Host generated TRANSFER command is successfully acknowledged by the TSC.	QUEUED		
2	QUEUED	The TRANSFER command has been initiated by the TSC.	WAITING	S6F11 Transfer-Initiated	Transport vehicle is dispatched to acquire the transfer unit.
3	WAITING	The acquire of the first carrier of the transfer unit begins.	TRANSFERRING	S6F11 Transferring	
4	NOT ACTIVE	Host issued cancel for the TRANSFER command is accepted by the TSC.	CANCELING	S6F11 TransferCancel-Initiated	A CANCEL remote command was issued to terminate a TRANSFER command that has not entered the physical aspect of the command.
5	CANCELING	Transport system is unable to cancel the TRANSFER command.	Previous NOT ACTIVE sub-state	S6F11 TransferCancel-Failed	The ability of the equipment to successfully complete a cancel of the TRANSFER command is specific to the IBSEM equipment supplier.
6	CANCELING	The cancel procedure for the TRANSFER command has completed by the transport system and TSC.	None	S6F11 TransferCancel-Completed	The transfer unit will still be situated at the transfer source location. The carriers in the transfer unit may now be included in a future transfer (either AMHS or PGV based).
7	TRANSFERRING	The TSC pauses execution of the TRANSFER command due to an anomaly condition.	PAUSED	S6F11 TransferPaused	It is an important distinction to make that the TRANSFER command is paused even though the vehicle may not be.
8	PAUSED	The TSC resumes execution of the TRANSFER command since the anomaly condition has been cleared.	TRANSFERRING	S6F11 Transfer-Resumed	
9	ACTIVE	Host initiates an abort of a TRANSFER command.	ABORTING	S6F11 TransferAbort-Initiated	An ABORT remote command was issued to terminate a TRANSFER command.
10	ABORTING	The abort procedure for the TRANSFER command has completed by the transport system and TSC.	None	S6F11 TransferAbort-Completed	Transfer unit could be located at any location or port located along the path of the ACTIVE transfer. The location of the carrier(s) associated with the aborted transfer command must be legal SourcePort(s) for issuing a new TRANSFER command.

Transition #	*Previous State*	*Trigger*	*New State*	*Actions*	*Comments*
11	ACTIVE	The TRANSFER command has completed by the transport system and TSC (either successfully or unsuccessfully).	None	S6F11 TransferCompleted sent to Host with appropriate ResultCode ResultCode = 0 if successful ResultCode Not = 0 if unsuccessful	Carrier(s) could be located at any location or port located along the path of the transfer if the TRANSFER command completed unsuccessfully. The location of the carrier(s) associated with an unsuccessful transfer command must be legal SourcePort(s) for a new TRANSFER command.
12	ABORTING	Transport system is physically unable to abort the TRANSFER command.	Previous ACTIVE sub-state	S6F11 TransferAbort-Failed	The ability of the equipment to successfully complete an ABORT of the TRANSFER command is specific to the IBSEM equipment supplier.
13	CANCELING	Transport system is unable to cancel the TRANSFER command because the transfer is now ACTIVE.	TRANSFERRING	S6F11 Transferring	The ability of the equipment to successfully complete a cancel of the TRANSFER command is specific to the IBSEM equipment supplier.

7.4 *Vehicle State Model*

7.4.1 *Vehicle State Model Requirements*

7.4.1.1 The purpose of the vehicle state model is to provide information to the host for use of transport vehicle information and metric tracking (i.e., the Host will not control vehicles). Each vehicle must individually comply with the vehicle state model. Implementation of this state model, along with associated events and variables, is not a requirement for Transport Systems which do not have vehicles. An example of such a Transport System is a DWC. The Host should not be dependent on any events from the vehicle state model.

7.4.2 *Vehicle State Model*

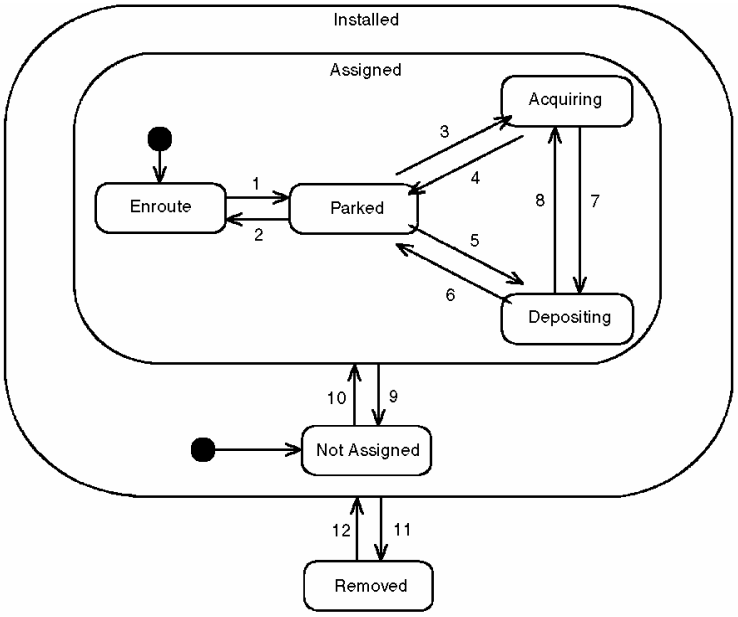

Figure 4
Generic IBSEM Vehicle State Model Diagram

7.4.3 *Vehicle State Definitions*

7.4.3.1 *INSTALLED* — The vehicle is available or being used for TRANSFER commands. All enabled collection events and alarms will be sent to the Host for vehicles in this state.

7.4.3.2 *REMOVED* — The vehicle is not available for Host initiated TRANSFER commands. No collection events or alarms will be sent to the Host for vehicles in this state.

7.4.3.3 *ASSIGNED (INSTALLED sub-state)* — Vehicle is allocated to a TRANSFER command.

7.4.3.4 *NOT ASSIGNED (INSTALLED sub-state)* — Vehicle is not allocated to a TRANSFER command. The vehicle may contain a carrier as the result of a command being aborted.

7.4.3.5 *ENROUTE (ASSIGNED sub-state)* — The vehicle is on its way to a transfer port. This is the default entry into the ASSIGNED state since it must be entered for the host to track vehicle metrics completely and adequately.

7.4.3.6 *PARKED (ASSIGNED sub-state)* — This state occurs when the vehicle is in the following conditions:

- After the arrival of the vehicle is completed and before the action of the transfer agent is started.
- After the action of the transfer agent has completed and before the departure of the vehicle.
- After continuous actions of the transfer agent (e.g., acquire/acquire and deposit/deposit) are completed.

7.4.3.7 *ACQUIRING (ASSIGNED sub-state)* — The vehicle is currently involved in carrier acquire (one or more carriers possible depending on vehicle limitations).

7.4.3.8 *DEPOSITING (ASSIGNED sub-state)* — The vehicle is currently involved in carrier deposit (one or more carriers possible depending on vehicle limitations).

7.4.4 *Vehicle State Transition Table*

Table 3 Vehicle State Transition Table

Transition #	*Previous State*	*Trigger*	*New State*	*Actions*	*Comment*
1	ENROUTE	Vehicle arrives at a transfer port associated with an ACTIVE transfer command.	PARKED	S6F11 VehicleArrived	TransferPort may be an *internal transfer port.*
2	PARKED	Vehicle departs a transfer port associated with an ACTIVE transfer command.	ENROUTE	S6F11 VehicleDeparted	TransferPort may be an *internal transfer port.*
3	PARKED	The carrier handoff parallel I/O starts for the vehicle to acquire (load) the transfer unit.	ACQUIRING	S6F11 VehicleAcquire-Started	If the vehicle is a passive type then the acquire occurs by the robot on the other equipment loading the transfer unit to the vehicle.
4	ACQUIRING	The carrier handoff parallel I/O completes for the vehicle to acquire (load) the transfer unit.	PARKED	S6F11 VehicleAcquire-Completed	
5	PARKED	The carrier handoff parallel I/O starts for the vehicle to deposit (unload) the transfer unit.	DEPOSITING	S6F11 VehicleDeposit-Started	If the vehicle is a passive type then the deposit occurs by the robot on the other equipment unloading the transfer unit from the vehicle.
6	DEPOSITING	The carrier handoff parallel I/O completes for the vehicle to deposit (unload) the transfer unit.	PARKED	S6F11 VehicleDeposit-Completed	

Transition #	*Previous State*	*Trigger*	*New State*	*Actions*	*Comment*
7	ACQUIRING	The carrier handoff parallel I/O completes for the vehicle to acquire (load) the transfer unit and starts for the vehicle to deposit (unload) the carrier.	DEPOSITING	S6F11 VehicleDeposit-Started	Carrier Replace See scenario for an example.
8	DEPOSITING	The carrier handoff parallel I/O completes for the vehicle to deposit (unload) the transfer unit and starts for the vehicle to acquire (load) the carrier.	ACQUIRING	S6F11 VehicleAcquire-Started	Carrier Replace See scenario for an example.
9	ASSIGNED	Vehicle is no longer being utilized for the specified command.	NOT ASSIGNED	S6F11 Vehicle-Unassigned	This could be the result of the command being completed or aborted. It could also be the result of the TSC scheduling algorithms assigning this vehicle to another command and/or another vehicle being assigned to this command.
10	NOT ASSIGNED	Vehicle is allocated to a TRANSFER command.	ASSIGNED	S6F11 VehicleAssigned	
11	INSTALLED	Vehicle is removed from use of transfer commands.	REMOVED	S6F11 VehicleRemoved	
12	REMOVED	Vehicle is installed for use of transfer commands.	INSTALLED	S6F11 VehicleInstalled	

7.5 *IBSEM Carrier State Model*

7.5.1 *IBSEM Carrier State Model Requirements*

7.5.1.1 The purpose of the carrier state model is to provide information to the host regarding carrier tracking (the Host will not control carriers). Each carrier must comply with the carrier state model.

7.5.2 *IBSEM Carrier State Model*

Figure 5
Generic IBSEM Carrier State Model Diagram

7.5.3 *Carrier State Definitions*

7.5.3.1 *INSTALLED* — The carrier is in the physical domain of the TSC.

7.5.4 *Carrier State Transition Table*

Table 4 Carrier State Transition Table

Transition #	*Previous State*	*Trigger*	*New State*	*Actions*	*Comments*
1	None	The carrier enters the physical domain of the TSC.	INSTALLED	S6F11 CarrierInstalled	
2	INSTALLED	The carrier is removed from the physical domain of the TSC.	None	S6F11 CarrierRemoved	

7.6 *Port Transfer State Model*

7.6.1 *Port Transfer State Model Requirements*

7.6.1.1 The purpose of the port transfer state model is to provided information to the host for the use in accessing ports. This may permit the host and stocker to utilize ports that are in service while avoiding the use of ports that are not in service.

7.6.2 *Port Transfer State Model*

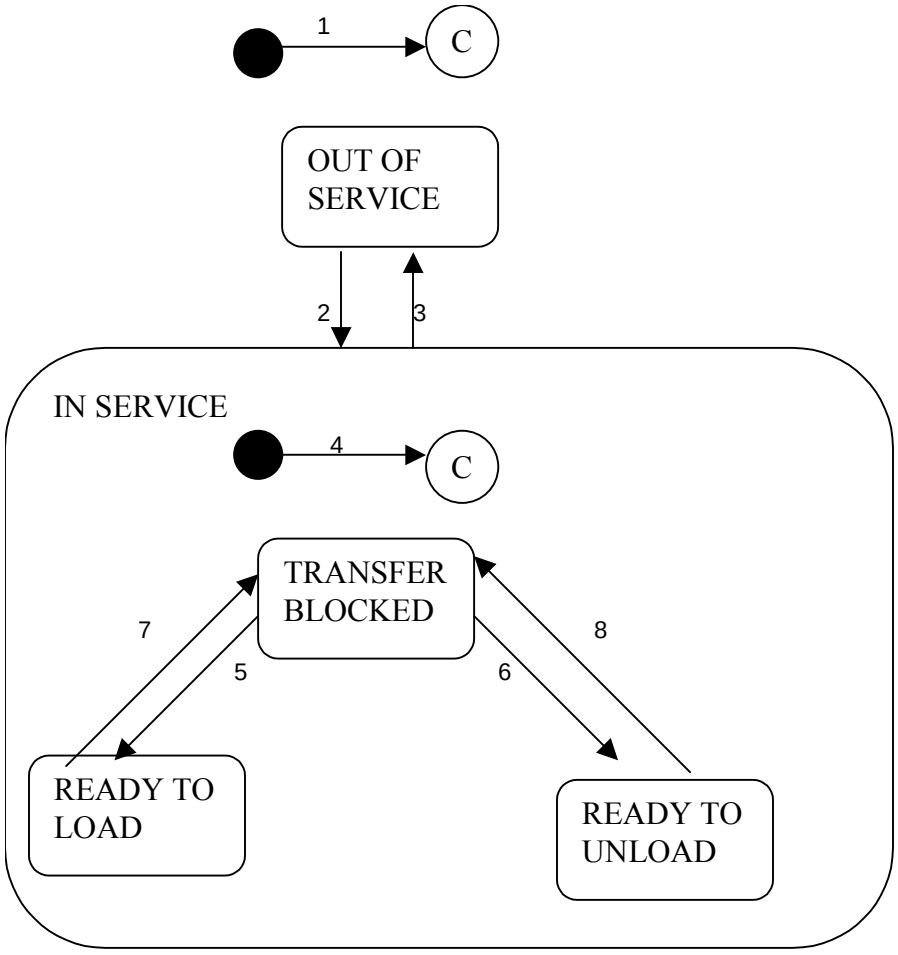

Figure 6
Port State Model Diagram

7.6.3 *Port Transfer State Definitions*

7.6.3.1 *OUT OF SERVICE* — Transfer to/from this port is disabled and the port should not be used in any Transfer command issued by the host. If a command is issued by the host which uses this port, it will not be rejected simply because the port is in this state.

7.6.3.2 *IN SERVICE* — Transfer to/from this port is enabled.

7.6.3.3 The following are **optional** sub-states of the IN SERVICE state to provide information to the host for the use in accessing ports. The information is generally obtained by the TSC via a low-level interface with the connected equipment. The connected equipment should generally, though not required, follow the Load Port Transfer State Model defined in SEMI E87. If these states are implemented, they are **not required** for every port.

7.6.3.3.1 *TRANSFER BLOCKED* — The state is neither READY TO LOAD nor READY TO UNLOAD.

7.6.3.3.2 *READY TO LOAD* — The port is able to accept (be loaded with) a carrier from the IBSEM equipment.

7.6.3.3.3 *READY TO UNLOAD* — The port is able to have a carrier removed (unloaded) by the IBSEM equipment.

7.6.4 *Port Transfer State Transition Table*

Table 5 Port Transfer State Transition Table

Transition #	*Previous State*	*Trigger*	*New State*	*Action*	*Comments*
1	None	System reset.	OUT OF SERVICE Or IN SERVICE	S6F11 PortOutOfService Or S6F11 PortInService	The new state is based on the current status of the port or the state prior to system reset.
2	OUT OF SERVICE	The equipment has determined that the port can be utilized for transfers.	IN SERVICE	S6F11 PortInService	
3	IN SERVICE	The equipment has determined that the port should not be used for transfers.	OUT OF SERVICE	S6F11 PortOutOfService	This could be the result of an alarm condition.
4	None	System Reset.	TRANSFER BLOCKED READY TO LOAD READY TO UNLOAD	S6F11 PortTransferBlocked S6F11 PortReadyToLoad S6F11 PortReadyToUnload	The new state is based on the current status of the port.
5	TRANSFER BLOCKED	Port ready for carrier delivery.	READY TO LOAD	S6F11 PortReadyToLoad	
6	TRANSFER BLOCKED	Port ready for carrier removal.	READY TO UNLOAD	S6F11 PortReadyToUnload	
7	READY TO LOAD	No carrier transfer allowed.	TRANSFER BLOCKED	S6F11 PortTransferBlocked	
8	READY TO UNLOAD	No carrier transfer allowed.	TRANSFER BLOCKED	S6F11 PortTransferBlocked	

8 Collection Event List

8.1 This section identifies data collection events and defines (Stream 6) suggested associated variable data items. The host can use the report definition scenario defined in SEMI E30 to define reports at IBSEM defined levels. The intent of this section is to demonstrate that certain suggested data is available at specific events.

8.2 *Requirements*

8.2.1 This standard requires all collection events listed in the SEMI E30 standard (according to the GEM capabilities required per §13).

8.2.2 *Collection Event Table*

Table 6 Collection Event Table (State Transition Based)

Collection Event Name	*From State*	*To State*	*Required DVVALs*
TSC STATE TRANSITION EVENTS			
AlarmCleared	ALARMS	NO ALARMS	CommandID VehicleInfo
AlarmSet	NO ALARMS ALARMS	ALARMS ALARMS	CommandID VehicleInfo
TSCAutoCompleted	PAUSED PAUSING	AUTO AUTO	N/A
TSCAutoInitiated	None	TSC INIT	N/A
TSCPauseCompleted	PAUSING	PAUSED	N/A
TSCPaused	TSC INIT	PAUSED	N/A
TSCPauseInitiated	AUTO	PAUSING	N/A
TRANSFER COMMAND STATE TRANSITION EVENTS			
TransferAbortCompleted	ABORTING	None	CommandID TransferCompleteInfo
TransferAbortFailed	ABORTING	ACTIVE (History)	CommandID
TransferAbortInitiated	ACTIVE	ABORTING	CommandID
TransferCancelCompleted	CANCELING	None	CommandID
TransferCancelFailed	CANCELING	NOT ACTIVE (History)	CommandID
TransferCancelInitiated	NOT ACTIVE	CANCELING	CommandID
TransferCompleted	ACTIVE	None	CommandInfo TransferCompleteInfo ResultCode
TransferInitiated	QUEUED	WAITING	CommandID
TransferPaused	TRANSFERRING	PAUSED	CommandID
TransferResumed	PAUSED	TRANSFERRING	CommandID
Transferring	WAITING CANCELING	TRANSFERRING TRANSFERRING	CommandID
VEHICLE STATE TRANSITION EVENTS			
VehicleArrived	ENROUTE	PARKED	VehicleID TransferPortList
VehicleAcquireStarted	PARKED DEPOSITING	ACQUIRING ACQUIRING	VehicleID TransferPort CarrierID (If Multi-position vehicles) TransferPortList CarrierIDList
VehicleAcquireCompleted	ACQUIRING	PARKED	VehicleID TransferPort CarrierID (If Multi-position vehicles) TransferPortList CarrierIDList
VehicleAssigned	NOT ASSIGNED	ASSIGNED	VehicleID CommandID
VehicleDeparted	PARKED	ENROUTE	VehicleID TransferPortList

Collection Event Name	*From State*	*To State*	*Required DVVALs*
VehicleDepositStarted	PARKED ACQUIRING	DEPOSITING DEPOSITING	VehicleID TransferPort CarrierID (If Multi-position vehicles) TransferPortList CarrierIDList
VehicleDepositCompleted	DEPOSITING	PARKED	VehicleID TransferPort CarrierID (If Multi-position vehicles) TransferPortList CarrierIDList
VehicleInstalled	REMOVED	INSTALLED	VehicleID
VehicleRemoved	INSTALLED	REMOVED	VehicleID
VehicleUnassigned	ASSIGNED	NOT ASSIGNED	VehicleID CommandID
CARRIER STATE TRANSITION EVENTS			
CarrierInstalled	None	INSTALLED	VehicleID CarrierID CarrierLoc CommandID
CarrierRemoved	INSTALLED	None	VehicleID CarrierID CarrierLoc CommandID
PORT TRANSFER STATE TRANSITION EVENTS			
PortInService	None OUT OF SERVICE	IN SERVICE	PortID
PortOutOfService	None IN SERVICE	OUT OF SERVICE	PortID
PortTransferBlocked	Any	TRANSFER BLOCKED	PortID
PortReadyToLoad	Any	READY TO LOAD	PortID
PortReadyToUnload	Any	READY TO UNLOAD	PortID

8.3 *Non-Transition Collection Event Table*

Table 7 Non-Transition Collection Event Table

Collection Event Name	*Event Description*	*Required DVVALs*
OperatorInitiatedAction	The operator initiated an action from the Transport System Controller. The related State Transition Events defined in Table 6 shall be required after this "OperatorInitiatedAction" event.	CommandID CommandType CarrierID SourcePort DestPort Priority
UnitStatusCleared	An Informational Status has been cleared on a 'transport unit'. This event is not required for compliance with the specification but shall be generated for every UnitStatusOccurred event that is generated with a UnitStatusClearable value of 'Y'.	UnitID, UnitStatusID, UnitStatusText, VehicleState, VehicleLocation The Vehicle variables are only required if a vehicle based system is being implemented.

Collection Event Name	*Event Description*	*Required DVVALs*
UnitStatusOccurred	An Informational Status occurred on a 'transport unit'. The supplier determines when and what 'status' information is to be supplied to the host. This event is not required for compliance with the specification.	UnitID, UnitStatusID, UnitStatusText, UnitStatusClearable, VehicleState, VehicleLocation The Vehicle variables are only required if a vehicle based system is being implemented.

9 Variable Data Items

9.1 The purpose of this section is to define the list of variable data item requirements for IBSEM equipment. Values of these variables will be available to the host via collection event reports and host status queries.

9.2 *Requirements*

- All variable data items defined in GEM and data item restrictions defined in SEMI E30 are required on IBSEM equipment (according to the GEM capabilities required per §13).
- All variable data items in the IBSEM Variable Data Item Dictionary for specific equipment classifications are required for IBSEM equipment. The data item restrictions are also required.
- Some SVs in the Variable Data Item Dictionary are referenced by an "i" subscript (e.g., $CarrierID_i$). The "i" subscript denotes a specific instance of the SV. This is necessary since there is usually more than one instance of such an SV active in the system at the same time (e.g., if there are 20 carriers active at the same time then "i" could range from 1 to 20 for $CarrierID_i$). Variable Data Items containing the "i" subscript should not have Variable IDs assigned to them.
- All variable data items with a format of ASCII (A) shall be limited to printable characters in the decimal range of 32 to 126 with the exception of the following non-permitted characters: "*" (decimal 42), "\"(decimal 92).

9.2.1 Variable data items are documented in the IBSEM Variable Data Item Dictionary using the following format:

Variable Name	*Type*	*Description*	*Class*	*Format*	*Comment*

Where:

Variable Name: A unique name for the variable data item.

Type: CV – meaning common variables, variables that are general to all vehicles.

CSV – meaning configuration specific variables.

Description: If class is DVVAL, then the description shall contain a statement of when data is valid in terms of IBSEM events.

Class: The data type of the item.

Format: < SECS Message Language (SML) mnemonic > acceptable formats are SEMI E5 lists, ASCII, floating point, unsigned integer or signed integer. A description of "ANY", indicates that only the above formats are acceptable and is left to the supplier to decide.

Comments: Any additional information pertinent to the variable name.

9.3 *Variable Data Item Types*

9.3.1 *Equipment Constants (ECV)* — The value can be changed by the host using S2F15. The operator may have the ability to change some or all of the values. The value of equipment constant may be queried at any time by the host using the S2F13/14 transaction or Stream 6 reports.

9.3.2 *Status Variables (SV)* — The values are valid at all times. A SV may not be changed by the host or operator, but may be changed by the equipment. A host or operator command may change an equipment status thus changing a SV. The value of status variables may be queried by the host at any time using the S1F3/4 or Stream 6 reports.

9.3.3 *Data Variables (DVVAL)* — These are variables which are valid upon the occurrence of a specific collection event, and may or may not be valid at other times depending upon the equipment. An attempt to read a variable data item when it's invalid will not result in an error, but the data reported may not have relevant meaning.

9.3.4 *Variable Data (V)* — This is a class of variable data which includes all the previously defined types of variables.

9.4 *Variable Data Item Dictionary*

Table 8 Variable Data Item Dictionary

Variable Name	*Type*	*Description*	*Class*	*Format*	*Comment*
ActiveCarriers	CV	List current status of all INSTALLED carriers.	SV	L,n 1. <CarrierInfo$_1$> . . n. <CarrierInfo$_n$>	
ActiveTransfers	CV	List current status of all ACTIVE TRANSFER commands.	SV	L,n 1. <TransferCommand$_1$> . . n. <TransferCommand$_n$>	
ActiveVehicles	CV	List current status of all vehicles available or being used for TRANSFER commands.	SV	L,n 1. <VehicleInfo$_1$> . . n. <VehicleInfo$_n$>	
CarrierID	CV	ID of the carrier being moved. CarrierID must be unique for each carrier within the TSC.	DVVAL	A[1–64]	If an Id is created by the equipment (not obtained via an id reader, the host interface, or the user interface) it must be of the following format: UNKNOWNEqpNameSeq Where: UNKNOWN are the exact characters "UNKNOWN" EqpName is the value of the EqpName ECV (truncated if required) Seq is a unique sequence identifier determined by the vendor.
CarrierID$_i$	CV	ID of the ith carrier.	SV	A[1–64]	See comment for CarrierID.
CarrierIDList	CV	The Ids of the Carriers being moved.	DVVAL	L,n 1.<CarrierID$_1$> . . n.<CarrierID$_n$>	'n' is the number of carriers being simultaneously transferred.
CarrierInfo	CV	All database information associated with a particular carrier generating an event.	DVVAL	L,3 1. <CarrierID> 2. <VehicleID> 3. <CarrierLoc>	

Variable Name	*Type*	*Description*	*Class*	*Format*	*Comment*
$CarrierInfo_i$	CV	All database information associated with the i^{th} carrier.	SV	L,3 1. <$CarrierID_i$> 2. <$VehicleID_i$> 3. <$CarrierLoc_i$>	
CarrierLoc	CSV	Unique location of the carrier within ITS as reported by the TSC.	DVVAL	A[1–64]	For multiple position vehicles, the "CarrierLoc" must be unique for each position on the vehicle and must be distinct from a location on any other vehicle.
$CarrierLoc_i$	CSV	Unique Location of the ith carrier within ITS as reported by the TSC.	SV	A[1–64]	
CommandID	CV	Remote Command ID Command ID generated by TSC.	DVVAL	A[1–64]	Used to subsequently refer to a specified remote command (e.g., to cancel a remote command). If a command is generated by the Transport System Controller using Non-Transition Collection Event "OperatorInitiatedAction", the commandId must begin with the string 'MANUAL' followed by any arbitrary sequence identifier.
$CommandID_i$	CV	The i^{th} Remote Command ID. The i^{th} Command ID generated by TSC.	SV	A[1–64]	Used to subsequently refer to a specified remote command (e.g., to cancel a remote command).
CommandInfo	CV	Command information associated with a particular transfer command.	DVVAL	L,3 1. <CommandID> 2. <Priority> 3. <Replace>	
$CommandInfo_i$	CV	Command information associated with the i^{th} transfer command.	SV	L,3 1. <$CommandID_i$> 2. <$Priority_i$> 3. <$Replace_i$>	
CommandName	CV	Host command issued to controller.	DVVAL	A[1–20]	
CommandType	CV	The type of Command being initiated.	DVVAL	A[1-20]	Valid Values are 'TRANSFER' 'CANCEL' 'ABORT'
CurrentPortStates	CV	Current State of all the ports.	SV	L,n 1. <$PortInfo_1$> . . n. <$PortInfo_n$>	
DestPort	CV	Destination port unique identifier.	DVVAL	A[1–64]	Must be the name of the port not the corresponding transport system node name or number.

Variable Name	*Type*	*Description*	*Class*	*Format*	*Comment*
$DestPort_i$	CV	The i^{th} Destination port unique identifier.	SV	A[1–64]	Must be the name of the port not the corresponding transport system node name or number.
EnhancedCarriers	CV	List Current status of all carrier information in the TSC database. This includes all carriers for which there are Transfer commands.	SV	L,n 1. <EnhancedCarrierInfo$_1$> . . n. <EnhancedCarrierInfo$_n$>	
EnhancedCarrier-Info$_i$	CV	All database information associated with a particular carrier.	SV	L,4 1. <CarrierID$_i$> 2. <VehicleID$_i$> 3. <CarrierLoc$_i$> 4. <InstallTime$_i$>	
EnhancedTransfers	CV	List current status of ALL transfer commands.	SV	L,n 1. <EnhancedTransferCommand$_1$> . . n. <EnhancedTransferCommand$_n$>	
EnhancedTransfer Command$_i$	CV	Information associated with a particular Transfer command.	SV	L,3 1. <CommandInfo$_i$> 2. <TransferState$_i$> 3. L,n 1. <TransferInfo$_1$> . n. <TransferInfo$_n$>	
EnhancedVehicles	CV	List current status of all vehicles available or being used for TRANSFER commands.	SV	L,n 1. <EnhancedVehicle$_1$Info> . . n.<EnhancedVehicleInfo$_n$>	
EnhancedVehicle-Info$_i$	CV	Information associated with a particular vehicle.	SV	L,3 1. <VehicleID> 2. <VehicleState> 3. <VehicleLocation>	
EqpName	CV	Unique ID of the TSC.	ECV	A[1–80]	Like a device name.
$InstallTime_i$	CV	Time the carrier was created in the TSC database.	SV	TIME (A16)	yyyymmddhhmmsscc
PortID	CV	ID of the port.	DVVAL	A[1–64]	
$PortID_i$	CV	ID of the port.	SV	A[1–64]	
$PortInfo_i$	CV	Port information associated with a particular port.	SV	L,2 1. <PortID$_i$> 2. <PortTransferState$_i$>	
$PortTransferState_i$	CV	Port Transfer State.	SV	U2	1 – OutOfService 2 – InService 3 – TransferBlocked 4 – ReadyToLoad 5 – ReadyToUnload

Variable Name	*Type*	*Description*	*Class*	*Format*	*Comment*
Priority	CV	Remote command priority.	DVVAL	U2	0 is not valid. 1 is the LOWEST priority, 99 is the highest priority.
$Priority_i$	CV	The i^{th} Remote command priority.	SV	U2	0 is not valid. 1 is the LOWEST priority, 99 is the highest priority.
Replace	CV	Flag to denote if a transfer command involves a carrier replace at the DestPort.	DVVAL	U2	0 = OFF > 0 = ON
$Replace_i$	CV	The i^{th} flag used to denote if a transfer command involves a carrier replace at the DestPort.	SV	U2	0 = OFF > 0 = ON
ResultCode	CV	Result Code of a transport system command. Associated with the command completion event.	DVVAL	U2 Successful = 0 Unsuccessful ≠ 0 ResultCodes that must be implemented are: Canceled Aborted	Values of ResultCode will correspond to meaningful completion results (0 always signifies normal successful completion).
SourcePort	CV	Source port unique identifier.	DVVAL	A[1–64]	Must be the name of the port not the corresponding transport system node name or number.
$SourcePort_i$	CV	The i^{th} Source port unique identifier.	SV	A[1–64]	Must be the name of the port not the corresponding transport system node name or number.
SpecVersion	CV	Version of SEMI E82 to which the equipment is compliant.	SV	A[0–20]	Example values are: SEMI E82-0999, SEMI E82-0301. If the equipment is not compliant, a zero length value may be specified.
TransferCommand	CV	Information associated with a particular TRANSFER command.	DVVAL	L,n 1. <CommandInfo> 2. <$TransferInfo_1$> . . . n. <$TransferInfo_m$>	m ≤ Number of carriers in the Transfer Unit
$TransferCommand_i$	CV	Information associated with the i^{th} TRANSFER command.	SV	L,2 1. <$CommandInfo_i$> 2. L, m 1. <$TransferInfo_1$> . . . m. <$TransferInfo_m$>	m = Number of carriers in the Transfer Unit

Variable Name	*Type*	*Description*	*Class*	*Format*	*Comment*
TransferComplete-Info	CV	Carrier information associated with a transfer.	DVVAL	L,n 1.<L,2 1. <TransferInfo$_1$> 2. <CarrierLoc$_1^{\#1}$> : : n.<L,2> 1. <TransferInfo$_n$> 2. <CarrierLoc$_n^{\#1}$>	N : size of Transfer Unit
TransferInfo	CV	Carrier information associated with a particular transfer command.	DVVAL	L,3 1. <CarrierID> 2. <SourcePort> 3. <DestPort>	
TransferInfo$_i$	CV	Carrier information associated with the ith transfer command.	SV	L,3 1. <CarrierID$_i$> 2. <SourcePort$_i$> 3. <DestPort$_i$>	
TransferPort	CV	Transfer Port unique identifier.	DVVAL	A[1–64]	Must be the name of the port where the transfer is taking place, not the corresponding transport system node name or number. May be the unique name of an *internal transfer port,* and therefore may not correspond to a valid SourcePort or DestPort value.
TransferPortList	CV	Transfer Port information associated with a particular vehicle arrival or departure event.	DVVAL	L,n 1. <TransferPort$_1$> 2. <TransferPort$_2$> . . . n. <TransferPort$_n$>	n > 1 for simultaneous transfers
TransferState$_i$	CV	State of Transfer Command.	SV	U2	1. Queued 2. Transferring 3. Paused 4. Canceling 5. Aborting 6. Waiting
TSCState	CV	TSC State (SYSTEM).	SV	U2	1 = TSC Init 2 = Paused 3 = Auto 4 = Pausing
UnitID	CV	The Id of the ‘transport unit’.	DVVAL	A[1–64]	This is the id of a port, vehicle, etc.
UnitLocation	CV	The location of the ‘transport unit’.	DVVAL	A[0-64]	This is the id of the location of the ‘transport unit’. It could be the id of a port, a section of track, etc.

Variable Name	*Type*	*Description*	*Class*	*Format*	*Comment*
UnitStatus Clearable	CV	When a UnitStatusOccured event is generated this flag indicates if the equipment is going to generate a UnitStatusCleared event when the status is corrected.	DVVAL	A[1]	'Y' – A UnitStatusCleared event will be generated. 'N' – A UnitStatusCleared event will be NOT be generated.
UnitStatusID	CV	The identification of the status.	DVVAL	A[1–64]	This is generally a vendor determined id which identifies the status. For example it could be "LostCommunication" or "SensorMalfunction" or "1234".
UnitStatusText	CV	Vendor specific information relating to the UnitStatusID	DVVAL	A[0-80]	Information that may be helpful to an operator. This could include more precise location information or other related information.
VehicleID	CV	Unique identification of a vehicle associated with an event.	DVVAL	A[1–32]	
$VehicleID_i$	CV	Unique identification of the ith vehicle.	SV	A[1–32]	
VehicleInfo	CV	Information associated with a particular vehicle.	DVVAL	L,2 1. <VehicleID> 2. <VehicleState>	
$VehicleInfo_i$	CV	Information associated with the ith vehicle.	SV	L,2 1. <$VehicleID_i$> 2. <$VehicleState_i$>	
VehicleLocation	CV	Location of the vehicle.	DVVAL	A[0–64]	The vehicle's port location. The data is only valid if the vehicle is Parked, Acquiring, or Depositing.
$VehicleLocation_i$	CV	Location of the vehicle.	SV	A[0–64]	The vehicle's port location. The data is only valid of the vehicle is Parked, Acquiring, or Depositiong.
VehicleState	CV	The state of the vehicle.	DVVAL	U2	1 = Removed 2 = Not Assigned 3 = Enroute 4 = Parked 5 = Acquiring 6 = Depositing
$VehicleState_i$	CV	The state of the ith vehicle.	SV	U2	See VehicleState above.

[#1] Current location (port or vehicle id) of the carrier is reported in 'CarrierLoc'. This may be used as a source port in a following transfer command.

10 Alarm List

10.1 Since each model of IBSEM equipment differs in configuration, it is not practical to provide an exhaustive list of all possible alarms. Instead, the IBSEM is requiring the two tables provided as described in SEMI E30 (Document Section). Alarm List Table which is intended to provide for equipment configuration specific alarms and Alarm ID, Alarm Set/Cleared Event Table. Any alarm that is displayed locally at the equipment, if enabled, is required to be sent to the host. To be compliant, Tables 9 and 10 must be completed by the supplier, documenting all alarms.

10.2 *Alarm List Table*

10.2.1 The alarm list table contains examples of alarms that pertain to various configuration aspects of equipment. These examples are intended to illustrate that alarms pertain to situations in which there exists a potential for exceeding physical safety limits associated with people, equipment, and material being transported as per the SEMI E30 definition of an alarm. See SEMI E30 for further reference. The supplier is responsible for supplying documentation associated with these alarm definitions. Each alarm will have an associated alarm text (ALTX) and alarm identifier (ALID). Table 9 contains example alarm list information that is intended to be augmented when the IBSEM equipment supplier documents its interface. Examples highlighted by (*) are required by IBSEM.

Table 9 Alarm List Table

Equipment Cfg.	*Alarm Text*	*ALID*	*Danger*		*Affected*		
			Potential	*Imminant*	*Operator*	*Equipment*	*Material*
OHT, OHS, RGT, AGT, and DWC	vehicle obstruction* (exceeded timeout)		X			X	
	transport system equipment failure*		X			X	X
	Carrier Handoff Parallel I/O failure*		X		X	X	X
	database error*		X			X	

10.3 *Alarm ID, Alarm Set/Cleared Event Table*

10.3.1 The Alarm ID, Alarm Set/Cleared Event table documents the association of each ALID to a set and cleared event as required by SEMI E30. See SEMI E30 for further reference. The supplier is responsible for supplying documentation associated with these alarm definitions. Each alarm will have associated alarm set and cleared collection event identifiers ($CEID_{set}$ and $CEID_{clear}$). Table 10 contains example alarm event information that is intended to be replaced when the IBSEM equipment supplier documents its interface.

Table 10 Alarm ID, Alarm Set/Cleared Event Table

Alarm ID (ALID)	*Alarm SET Event ($CEID_{set}$)*	*Alarm CLEARED Event ($CEID_{cleared}$)*

11 Remote Commands

11.1 The purpose of this section is to identify remote commands, command parameters, and valid commands versus states pertinent to the SEM. All remote commands identified in this section follow the format of the S2,F41 Host Command Send SECS-II message except for the TRANSFER and STAGE commands, which follow the S2,F49 Enhanced Remote Command Send SECS-II message.

11.2 *Requirements*

- The equipment shall support the SEMI E30 (according to the GEM capabilities required per §13) required remote commands.
- All the remote commands defined by IBSEM are required to be implemented as specified.
- The alphanumeric strings defined by IBSEM for RCMD and CPNAME are required.
- A completed table must be generated where an "X" is placed in the table for each state that a given command is valid.

- If additional remote commands are supported then a "remote commands versus valid states" matrix must be generated for these additional commands.
- For additional commands, a table must be generated similar to the remote command descriptions summary.

11.3 *Remote Commands Description*

11.3.1.1 *ABORT* — This command terminates the activity of a specific TRANSFER command based on CommandID while the command is in the ACTIVE state. This command may not be accepted due to mechanical issues if the vehicle is in a specific condition (e.g., depositing a carrier). The exact conditions surrounding when the ABORT command is not accepted by the TSC must be documented by the IBSEM equipment supplier.

11.3.1.2 *CANCEL* — This command terminates the activity of a specific TRANSFER command based on CommandID while the command is in either the QUEUED or WAITING state. This command must always be accepted by the TSC when in the QUEUED or WAITING state.

11.3.1.3 *PAUSE* — This command puts the TSC in the PAUSING state.

11.3.1.4 *RESUME* — This command puts the TSC in the AUTO state.

11.3.1.5 *TRANSFER* — This is a SECS-II Enhanced Remote Command instead of a SECS-II Host Command Send (S2,F49 instead of S2,F41). See the examples in Related Information 1 for details.

11.3.1.5.1 This command is used to perform the entire transfer command for the carrier(s) to be transferred between transfer ports. The execution of this command will include allocation of a vehicle, acquiring the carrier(s), moving the carrier(s) to the destination port(s), depositing the carrier(s), and returning the vehicle for other use. The number of carriers in the TRANSFER command is less than or equal to the number of carriers in the transfer unit (see ¶5.2 for definition of the transfer unit). It is recommended that the Carrier already be at the SOURCEPORT upon the issue of the TRANSFER command, otherwise, it is possible that the TRANSFER command can fail on "empty acquire" when no carrier is present on the SOURCEPORT when the acquire is started.

11.3.1.5.2 This command shall not be rejected or denied for the sole reason that a STAGEID in the STATEIDLIST parameter does not reference the STAGEID of any current STAGE command. The TRANSFER command shall be processed as if the invalid STAGEID was not present in the message.

11.3.1.6 *INFOUPDATE* — This is used to associate information with a carrier while in the TSC database. This command is an optional feature and is not required for compliance. If this command is not supported, then a HCACK of '1' (Command does not exist) shall be returned. If the carrier does not exist in the TSC database, then a HCACK of '3' (At least one parameter is invalid) shall be returned.

11.3.1.7 *STAGE* — This is used to inform the TSC that a transport will be requested in the future. This command allows, but does not require, the TSC to perform Transfer optimizations. This command is an optional feature and is not required for compliance. If this command is not supported, then a HCACK of '1' (Command does not exist) shall be returned. It is also permissible for the command to be accepted and no action taken. The Host is not required to send this command prior to a TRANSFER. Multiple valid STAGE commands for the same SOURCEPORT can exist at the same time. A STAGE command becomes invalid once the subsequent TRANSFER command is received, once the WAITTIMEOUT has expired, or once it has been deleted with STAGEDELETE. The use of this command may cause premature blocking of sections of the transport system based upon the values of the timers along with the TSC software and hardware implementations.

11.3.1.8 *STAGEDELETE* — This command is used to delete STAGE commands. If a STAGEID is specified, then that STAGE command shall be deleted. If the STAGEID is not specified, (STAGEID cpname parameter is not included in the message), then all STAGE commands shall be deleted. It can be used in anomaly situations when a TRANSFER command will not be sent, or prior to sending an updated STAGE command, or during recover scenarios.

11.3.2 *Remote Commands and Associated Host Command Parameters*

11.3.2.1 This table describes the allowable command parameters (CPNAME) for each remote command (RCMD). Equipment shall support all parameters. The column marked Req/Opt, specifies which parameters are required to be sent by the host and which parameters may be optionally sent by the host.

Table 11 Allowable Command Parameters

Remote Command	*Parameters*		
	Cpname	*Req/opt*	*Comment*
ABORT	"COMMANDID"	R	Must specify the commandID that was used for the TRANSFER command that is being ABORT'ed.
CANCEL	"COMMANDID"	R	Must specify the commandID that was used for the TRANSFER command that is being CANCEL'ed.
INFOUPDATE	"CARRIERID" *"HostDefined"*	R O	The *"HostDefined"* cpnames are used to indicate the name of the data that is to be associated with a carrier. See §12 for example scenarios. The data associated with a carrier is for display purposes only and information such as the carrier's current location and state cannot be changed with this command.
PAUSE	None	NA	Once received by the TSC, the TSC will queue any TRANSFER commands until the TSC receives and successfully executes the RESUME command. Once in the AUTO state the TSC will process the TRANSFER commands in its queue.
RESUME	None	N/A	Returns the PAUSEd TSC to the AUTO state.
STAGE	"STAGEINFO" "TRANSFERINFO"*	R R	*STAGE commands are allowed a maximum of |transfer unit| TRANSFERINFOs. Any information that is unknown to the Host at the time the STAGE is sent shall be left blank in the TransferInfo. In some scenarios, information in the STAGE command may differ from that in the subsequent TRANSFER command.
STAGEDELETE	"STAGEID"	O	The STAGEID of the corresponding STAGE command to be deleted.
TRANSFER	"COMMANDINFO" "TRANSFERINFO"* "STAGEIDLIST"	R R O	* TRANSFER commands are allowed a maximum of |transfer unit| TRANSFERINFOs. STAGEIDLIST contains the STAGEID of each STAGE command that this TRANSFER command fulfills.

11.3.3 *Host Command Parameters Name and Values*

Table 12 Host Command Parameters CPNAMES

Cpname	*Parameter Value*		
	Description	*Range*	*Format*
CARRIERID	ID of the carrier being moved.		A[1–64]
COMMANDID	Unique command identifier created by the Host.		A[1–64]
COMMANDINFO	L,3 COMMANDID PRIORITY REPLACE		L,3
DESTPORT	Destination port unique identifier.		A[1–64]
EXPECTEDDURATION	The amount of time, in seconds, the host expects between the STAGE command and the subsequent TRANSFER command.	0-999	U2

Cpname	*Parameter Value*		
	Description	*Range*	*Format*
NOBLOCKINGTIME	The amount of time, in seconds, following a STAGE command that movement past the SOURCEPORT should not be blocked by a transport unit. This time should not exceed the ExpectedDuration.	0–999	U2
PRIORITY	Remote command priority.	0 is not valid. 1 is the LOWEST priority, 99 is the highest priority.	U2
REPLACE	Flag to denote a TRANSFER replace carrier. Replace a carrier already on the port with the one on the vehicle. Refers to the subsequent TRANSFER command when used in a STAGE command.	0 is OFF > 0 is ON	U2
SOURCEPORT	Source port unique identifier.		A[1–64]
STAGEID	The unique STAGE command identifier created by the host.		A[1–64]
STAGEIDLIST	L,n 1. <$STAGEID_1$> 2. <$STAGEID_2$> . . . n. <$STAGEID_n$>		L,n The format shall follow format 'A' as defined for the S2F49 message in SEMI E5. (The CPNAME "STAGEID" is NOT included, only values). For Example: L,2 1. "STAGEID LIST" 2. L,n 1, "Stage 1 2. "Stage2" n, "StageN"
STAGEINFO	L,6 STAGEID PRIORITY REPLACE EXPECTEDDURATION NOBLOCKINGTIME WAITTIMEOUT		L,6
TRANSFERINFO	L,3 CARRIERID SOURCEPORT DESTPORT		L,3
WAITTIMEOUT	The amount of time, in seconds, after the SOURCEPORT is reached by a transport unit and the ExpectedDuration has elapsed that the STAGE command should be considered valid by the TSC. Following the WaitTimeout, the Controller should disregard the STAGE command.	0–999	U2

11.3.4 *Remote Commands versus TSC and TRANSFER Command States*

11.3.4.1 The following table indicates TSC and TRANSFER Command States where the remote commands are allowed. This is indicated with a "X" mark. Remote commands act independently of other state models (e.g.,

Vehicle States and Carrier States are independent from the IBSEM remote commands). "NA" (Not Applicable) means that States and Remote Commands have no direct relationship.

Table 13 Remote Commands versus TSC and TRANSFER Command States

	COMMAND							
	TRANSFER	RESUME	PAUSE	CANCEL	ABORT	INFOUPDATE	STAGE	STAGEDELETE
TSC STATE								
AUTO	X		X	X	X	X	X	X
ALARMS	X	X	X	X	X	X	X	X
NO ALARMS	X	X	X	X	X	X	X	X
INIT								
PAUSED	X	X		X	X	X	X	X
PAUSING	X	X		X	X	X	X	X
TRANSFER COMMAND STATE								
QUEUED	N/A	N/A	N/A	X		N/A	N/A	N/A
WAITING	N/A	N/A	N/A	X		N/A	N/A	N/A
ACTIVE (PAUSED or TRANS.)	N/A	N/A	N/A		X	N/A	N/A	N/A
ABORTING	N/A	N/A	N/A			N/A	N/A	N/A
CANCELING	N/A	N/A	N/A			N/A	N/A	N/A

12 Scenarios

12.1 The following scenarios represent Application Notes. In the scenarios, all unique Remote Command IDs must initially be created and sent by the Host. Subsequent event reports sent from the equipment referring to the status of a particular remote command must return the applicable CommandID. All collection events identified in Table 6 are assumed to be enabled (per the SEMI E30 definition/scenario) throughout the following scenarios. Variable data specified in the Host commands has been chosen arbitrarily for the purpose of demonstrating message structure/content. The Collection Event Report definitions contained in the scenarios are examples that could be defined by the Host.

12.2 *Normal Transport*

12.2.1 *Single Carrier Transfer — Transfer Unit Size is Equal to 1 Carrier*

12.2.1.1 A carrier is transported from a source port to a destination port.

STEP	COMMENTS	HOST	TSC	COMMENTS
1.	Carrier 123456 is sitting at PORTXX prepared for a TRANSFER command.			
2.	Enhanced Remote Command (ERC) **TRANSFER** . COMMANDID = "111111" . PRIORITY = 5 . REPLACE = 0 . $TRANSFERINFO_1$ - L,3 1. CARRIERID = "123456" 2. SOURCEPORT = "PORTXX" 3. DESTPORT = "PORTYY"	S2,F49->		

STEP	COMMENTS	HOST	TSC	COMMENTS
3.			<-S2,F50	Enhanced Remote Command Acknowledge (ERCA)
4.			<-S6,F11	Event Report Send (ERS) TransferInitiated . CommandID = "111111"
5.	Event Report Acknowledge (ERA)	S6,F12->		
6.			<-S6,F11	Event Report Send (ERS) VehicleAssigned . VehicleID = "CARXX" /* Actual VehicleID used for the transfer may be different due to TSC scheduling optimizations */ CommandID = "111111"
7.	Event Report Acknowledge (ERA)	S6,F12->		
8.			<-S6,F11	Event Report Send (ERS) VehicleArrived . VehicleID = "CARXX" . TransferPortList - L,1 1. TransferPort = "PORTXX"
9.	Event Report Acknowledge (ERA)	S6,F12->		
10.			<-S6,F11	Event Report Send (ERS) Transferring . CommandID = "111111"
11.	Event Report Acknowledge (ERA)	S6,F12->		
12.			<-S6,F11	Event Report Send (ERS) VehicleAcquireStarted . VehicleID = "CARXX" . TransferPort = "PORTXX" . CarrierID = "123456"
13.	Event Report Acknowledge (ERA)	S6,F12->		
14.			<-S6,F11	Event Report Send (ERS) CarrierInstalled . VehicleID = "CARXX" . CarrierID = "123456" . CarrierLoc = "LOC1"
15.	Event Report Acknowledge (ERA)	S6,F12->		
16.			<-S6,F11	Event Report Send (ERS) VehicleAcquireCompleted . VehicleID = "CARXX" . TransferPort = "PORTXX" . CarrierID = "123456"
17.	Event Report Acknowledge (ERA)	S6,F12->		

STEP	COMMENTS	HOST	TSC	COMMENTS
18.			<-S6,F11	Event Report Send (ERS) VehicleDeparted . VehicleID = "CARXX" . TransferPortList - L,1 1. TransferPort = "PORTXX"
19.	Event Report Acknowledge (ERA)	S6,F12->		
20.			<-S6,F11	Event Report Send (ERS) VehicleArrived . VehicleID = "CARXX" . TransferPortList - L,1 1. TransferPort = "PORTYY"
21.	Event Report Acknowledge (ERA)	S6,F12->		
22.			<-S6,F11	Event Report Send (ERS) VehicleDepositStarted . VehicleID = "CARXX" . TransferPort = "PORTYY" . CarrierID = "123456"
23.	Event Report Acknowledge (ERA)	S6,F12->		
24.			<-S6,F11	Event Report Send (ERS) CarrierRemoved . VehicleID = "CARXX" . CarrierID = "123456" . CarrierLoc = "LOC1"
25.	Event Report Acknowledge (ERA)	S6,F12->		
26.			<-S6,F11	Event Report Send (ERS) VehicleDeposit-Completed . VehicleID = "CARXX" . TransferPort = "PORTYY" . CarrierID = "123456"
27.	Event Report Acknowledge (ERA)	S6,F12->		
28.			<-S6,F11	Event Report Send (ERS) VehicleUnassigned . VehicleID = "CARXX" . CommandID = "111111"
29.	Event Report Acknowledge (ERA)	S6,F12->		

STEP	COMMENTS	HOST	TSC	COMMENTS
30.			<-S6,F11	Event Report Send (ERS) TransferCompleted . CommandName = "TRANSFER" . CommandID = "111111" . Priority = 5 . Replace = 0 . ResultCode = 0 . CarrierLoc = "PORTYY" . $TransferInfo_1$ - L,3 1. CarrierID = "123456" 2. SourcePort = "PORTXX" 3. DestPort = "PORTYY"
31.	Event Report Acknowledge (ERA)	S6,F12->		

12.2.2 *Simultaneous Multiple Carrier Transfer — Transfer Unit Size is Equal to 2 Carriers*

12.2.2.1 Two carriers are transported from 2 source ports (e.g., L-Shaped stocker output ports) to 2 destination ports (e.g., process equipment ports). The transport vehicle is capable of acquiring and depositing 2 carriers at the same time (e.g., forking type vehicle). It is assumed that the source and destination ports must be the same equipment.

STEP	COMMENTS	HOST	TSC	COMMENTS
1.	Both carriers are prepared for a TRANSFER command at the source ports.			
2.	Enhanced Remote Command (ERC) . TRANSFER . COMMANDID = "111111" . PRIORITY = 5 . REPLACE = 0 . $TRANSFERINFO_1$ - L,3 1. CARRIERID = "123456" 2. SOURCEPORT = "PORTX1" 3. DESTPORT = "PORTY1" . $TRANSFERINFO_2$ - L,3 4. CARRIERID = "654321" 5. SOURCEPORT = "PORTX2" 6. DESTPORT = "PORTY2"	S2,F49->		
3.			<-S2,F50	Enhanced Remote Command Acknowledge (ERCA)
4.			<-S6,F11	Event Report Send (ERS) TransferInitiated . CommandID = "111111"
5.	Event Report Acknowledge (ERA)	S6,F12->		
6.			<-S6,F11	Event Report Send (ERS) VehicleAssigned . VehicleID = "AGVXX" . CommandID = "111111"
7.	Event Report Acknowledge (ERA)	S6,F12->		

STEP	COMMENTS	HOST	TSC	COMMENTS
8.			<-S6,F11	Event Report Send (ERS) VehicleArrived . VehicleID = "AGVXX" . TransferPortList – L,2 1. TransferPort$_1$ = "PORTX1" 2. TransferPort$_2$ = "PORTX2"
9.	Event Report Acknowledge (ERA)	S6,F12->		
10.			<-S6,F11	Event Report Send (ERS) Transferring . CommandID = "111111"
11.	Event Report Acknowledge (ERA)	S6,F12->		
12.			<-S6,F11	Event Report Send (ERS) VehicleAcquireStarted . VehicleID = "AGVXX"
13.	Event Report Acknowledge (ERA)	S6,F12->		
14.			<-S6,F11	Event Report Send (ERS) CarrierInstalled . VehicleID = "AGVXX" . CarrierID = "123456" . CarrierLoc = "LOC1"
15.	Event Report Acknowledge (ERA)	S6,F12->		
16.			<-S6,F11	Event Report Send (ERS) CarrierInstalled . VehicleID = "AGVXX" . CarrierID = "654321" . CarrierLoc = "LOC2"
17.	Event Report Acknowledge (ERA)	S6,F12->		
18.			<-S6,F11	Event Report Send (ERS) VehicleAcquireCompleted . VehicleID = "AGVXX"
19.	Event Report Acknowledge (ERA)	S6,F12->		
20.			<-S6,F11	Event Report Send (ERS) VehicleDeparted . VehicleID = "AGVXX" . TransferPortList – L,2 1. TransferPort$_1$ = "PORTX1" 2. TransferPort$_2$ = "PORTX2"
21.	Event Report Acknowledge (ERA)	S6,F12->		
22.			<-S6,F11	Event Report Send (ERS) VehicleArrived . VehicleID = "AGVXX" . TransferPortList – L,2 1. TransferPort$_1$ = "PORTY1" 2. TransferPort$_2$ = "PORTY2"

STEP	COMMENTS	HOST	TSC	COMMENTS
23.	Event Report Acknowledge (ERA)	S6,F12->		
24.			<-S6,F11	Event Report Send (ERS) VehicleDepositStarted . VehicleID = "AGVXX"
25.	Event Report Acknowledge (ERA)	S6,F12->		
26.			<-S6,F11	Event Report Send (ERS) CarrierRemoved . VehicleID = "AGVXX" . CarrierID = "123456" . CarrierLoc = "LOC1"
27.	Event Report Acknowledge (ERA)	S6,F12->		
28.			<-S6,F11	Event Report Send (ERS) CarrierRemoved . VehicleID = "AGVXX" . CarrierID = "654321" . CarrierLoc = "LOC2"
29.	Event Report Acknowledge (ERA)	S6,F12->		
30.			<-S6,F11	Event Report Send (ERS) VehicleDeposit-Completed VehicleID = "AGVXX"
31.	Event Report Acknowledge (ERA)	S6,F12->		
32.			<-S6,F11	Event Report Send (ERS) VehicleUnassigned . VehicleID = "AGVXX" . CommandID = "111111"
33.	Event Report Acknowledge (ERA)	S6,F12->		
34.			<-S6,F11	Event Report Send (ERS) TransferCompleted (Complete parameters) . CommandName = "TRANSFER" . CommandID = "111111" . Priority = 5 . Replace = 0 . ResultCode = 0 . $CarrierLoc_1$ = "PORTY1" . $TransferInfo_1$ – L,3 1. CarrierID = "123456" 2. SourcePort = "PORTX1" 3. DestPort = "PORTY1" . $CarrierLoc_2$ = "PORTY2" . $TransferInfo_2$ – L,3 4. CarrierID = "654321" 5. SourcePort = "PORTX2" 6. DestPort = "PORTY2"
35.	Event Report Acknowledge (ERA)	S6,F12->		

12.2.3 *Continuous Multiple Carrier Transfer – Transfer Unit Size is Equal to 2 Carriers*

12.2.3.1 Two carriers are transported from a single source port to a single destination port. This scenario is used to demonstrate how the TSC would handle a continuous load/unload request. It also shows the sequence of events associated with the continuous case. It is assumed that the source and destination ports must be the same equipment.

STEP	COMMENTS	HOST	TSC	COMMENTS
1.	The carriers are prepared for a TRANSFER command. They are sitting sequentially at the single pickup point source output. 123456 is at the vehicle accessible source port for the first acquire of two. 654321 is sitting right behind it, and it will be shuttled out after 123456 is acquired by the vehicle. The vehicle will then acquire 654321 once in position. The depositing order depends on the IBSEM supplier's functional specification.			
2.	Enhanced Remote Command(ERC) **TRANSFER** . COMMANDID = "111111" . PRIORITY = 5 . REPLACE = 0 . $TRANSFERINFO_1$ – L,3 1. CARRIERID = "123456" 2. SOURCEPORT = "PORTXX" 3. DESTPORT = "PORTYY" . $TRANSFERINFO_2$ – L,3 4. CARRIERID = "654321" 5. SOURCEPORT = "PORTXX" 6. DESTPORT = "PORTYY"	S2,F49->		
3.			<-S2,F50	Enhanced Remote Command Acknowledge (ERCA)
4.			<-S6,F11	Event Report Send (ERS) TransferInitiated . CommandID = "111111"
5.	Event Report Acknowledge (ERA)	S6,F12->		
6.			<-S6,F11	Event Report Send (ERS) VehicleAssigned . VehicleID = "RGVXX" . CommandID = "111111"
7.	Event Report Acknowledge (ERA)	S6,F12->		
8.			<-S6,F11	Event Report Send (ERS) VehicleArrived . VehicleID = "RGVXX" . TransferPortList – L,1 1. TransferPort = "PORTXX"
9.	Event Report Acknowledge (ERA)	S6,F12->		

STEP	COMMENTS	HOST	TSC	COMMENTS
10.			<-S6,F11	Event Report Send (ERS) Transferring . CommandID = "111111"
11.	Event Report Acknowledge (ERA)	S6,F12->		
12.			<-S6,F11	Event Report Send (ERS) VehicleAcquireStarted . VehicleID = "RGVXX"
13.	Event Report Acknowledge (ERA)	S6,F12->		
14.			<-S6,F11	Event Report Send (ERS) CarrierInstalled . VehicleID = "RGVXX" . CarrierID = "123456" . CarrierLoc = "LOC1"
15.	Event Report Acknowledge (ERA)	S6,F12->		
16.			<-S6,F11	Event Report Send (ERS) CarrierInstalled . VehicleID = "RGVXX" . CarrierID = "654321" . CarrierLoc = "LOC2"
17.	Event Report Acknowledge (ERA)	S6,F12->		
18.			<-S6,F11	Event Report Send (ERS) VehicleAcquireCompleted . VehicleID = "RGVXX"
19.	Event Report Acknowledge (ERA)	S6,F12->		
20.			<-S6,F11	Event Report Send (ERS) VehicleDeparted . VehicleID = "RGVXX" . TransferPortList – L,1 1. TransferPort = "PORTXX"
21.	Event Report Acknowledge (ERA)	S6,F12->		
22.			<-S6,F11	Event Report Send (ERS) VehicleArrived . VehicleID = "RGVXX" . TransferPortList – L,1 1. TransferPort = "PORTYY"
23.	Event Report Acknowledge (ERA)	S6,F12->		
24.			<-S6,F11	Event Report Send (ERS) VehicleDepositStarted . VehicleID = "RGVXX"
25.	Event Report Acknowledge (ERA)	S6,F12->		
26.			<-S6,F11	Event Report Send (ERS) CarrierRemoved . VehicleID = "RGVXX" . CarrierID = "123456" . CarrierLoc = "LOC1"
27.	Event Report Acknowledge (ERA)	S6,F12->		

STEP	COMMENTS	HOST	TSC	COMMENTS
28.			<-S6,F11	Event Report Send (ERS) CarrierRemoved . VehicleID = "RGVXX" . CarrierID = "654321" . CarrierLoc = "LOC2"
29.	Event Report Acknowledge (ERA)	S6,F12->		
30.			<-S6,F11	Event Report Send (ERS) VehicleDeposit-Completed . VehicleID = "RGVXX"
31.	Event Report Acknowledge (ERA)	S6,F12->		
32.			<-S6,F11	Event Report Send (ERS) VehicleUnassigned . VehicleID = "RGVXX" . CommandID = "111111"
33.	Event Report Acknowledge (ERA)	S6,F12->		
34.			<-S6,F11	Event Report Send (ERS) TransferCompleted (Complete parameters) . CommandName = "TRANSFER" . CommandID = "111111" . Priority = 5 . Replace = 0 . ResultCode = 0 . $CarrierLoc_1$ = "PORTYY" . $TransferInfo_1$ – L,3 1. CarrierID = "123456" 2. SourcePort = "PORTXX" 3. DestPort = "PORTYY" . $CarrierLoc_2$ = "PORTYY" . $TransferInfo_2$ – L,3 4. CarrierID = "654321" 5. SourcePort = "PORTXX" 6. DestPort = "PORTYY"
35.	Event Report Acknowledge (ERA)	S6,F12->		

12.2.4 *Carriers Replace*

12.2.4.1 One carrier is transported from the source port (e.g., stocker output port) to the destination port (e.g., process equipment port-1) and the other carrier is transported from the same port (process equipment port-1) to the destination ports (e.g., stocker input port) by the same vehicle. The transport vehicle is capable of acquiring and depositing these two carriers sequentially. This function is used for the depressuring process equipment with door, because the door opening time should be shorten for this type of equipment.

STEP	COMMENTS	HOST	TSC	COMMENTS
1.	Both carriers are prepared for a TRANSFER command at the source ports.			

STEP	COMMENTS	HOST	TSC	COMMENTS
2.	Enhanced Remote Command (ERC) TRANSFER . COMMANDID = "111111" . PRIORITY = 5 . REPLACE = 1 . $TRANSFERINFO_1$ – L,3 1. CARRIERID = "123456" 2. SOURCEPORT = "PORTX1" 3. DESTPORT = "PORTY1" . $TRANSFERINFO_2$ – L,3 4. CARRIERID = "654321" 5. SOURCEPORT = "PORTY1" 6. DESTPORT = "PORTZ1"	S2,F49->		
3.			<-S2,F50	Enhanced Remote Command Acknowledge (ERCA)
4.			<-S6,F11	Event Report Send (ERS) TransferInitiated . CommandID = "111111"
5.	Event Report Acknowledge (ERA)	S6,F12->		
6.			<-S6,F11	Event Report Send (ERS) VehicleAssigned . VehicleID = "AGVXX" . CommandID = "111111"
7.	Event Report Acknowledge (ERA)	S6,F12->		
8.			<-S6,F11	Event Report Send (ERS) VehicleArrived . VehicleID = "AGVXX" . TransferPortList – L,1 1. $TransferPort_1$ = "PORTX1"
9.	Event Report Acknowledge (ERA)	S6,F12->		
10.			<-S6,F11	Event Report Send (ERS) Transferring . CommandID = "111111"
11.	Event Report Acknowledge (ERA)	S6,F12->		
12.			<-S6,F11	Event Report Send (ERS) VehicleAcquireStarted . VehicleID = "AGVXX"
13.	Event Report Acknowledge (ERA)	S6,F12->		
14.			<-S6,F11	Event Report Send (ERS) CarrierInstalled . VehicleID = "AGVXX" . CarrierID = "123456" . CarrierLoc = "LOC1"
15.	Event Report Acknowledge (ERA)	S6,F12->		
16.			<-S6,F11	Event Report Send (ERS) VehicleAcquireCompleted . VehicleID = "AGVXX"

STEP	COMMENTS	HOST	TSC	COMMENTS
17.	Event Report Acknowledge (ERA)	S6,F12->		
18.			<-S6,F11	Event Report Send (ERS) VehicleDeparted . VehicleID = "AGVXX" . TransferPortList – L,1 1. TransferPort$_1$ = "PORTX1"
19.	Event Report Acknowledge (ERA)	S6,F12->		
20.			<-S6,F11	Event Report Send (ERS) VehicleArrived . VehicleID = "AGVXX" . TransferPortList – L,1 1. TransferPort$_1$ = "PORTY1"
21.	Event Report Acknowledge (ERA)	S6,F12->		
22.			<-S6,F11	Event Report Send (ERS) VehicleAcquireStarted . VehicleID = "AGVXX"
23.	Event Report Acknowledge (ERA)	S6,F12->		
24.			<-S6,F11	Event Report Send (ERS) CarrierInstalled . VehicleID = "AGVXX" . CarrierID = "654321" . CarrierLoc = "LOC2"
25.	Event Report Acknowledge (ERA)	S6,F12->		
26.			<-S6,F11	Event Report Send (ERS) VehicleDepositStarted . VehicleID = "AGVXX"
27.	Event Report Acknowledge (ERA)	S6,F12->		
28.			<-S6,F11	Event Report Send (ERS) CarrierRemoved . VehicleID = "AGVXX" . CarrierID = "123456" . CarrierLoc = "LOC1"
29.	Event Report Acknowledge (ERA)	S6,F12->		
30.			<-S6,F11	Event Report Send (ERS) VehicleDepositCompleted . VehicleID = "AGVXX"
31.	Event Report Acknowledge (ERA)	S6,F12->		
32.			<-S6,F11	Event Report Send (ERS) VehicleDeparted . VehicleID = "AGVXX" . TransferPortList – L,1 1. TransferPort$_1$ = "PORTY1"
33.	Event Report Acknowledge (ERA)	S6,F12->		

STEP	COMMENTS	HOST	TSC	COMMENTS
34.			<-S6,F11	Event Report Send (ERS) VehicleArrived . VehicleID = "AGVXX" . TransferPortList – L,1 1. $TransferPort_1$ = "PORTZ1"
35.	Event Report Acknowledge (ERA)	S6,F12->		
36.			<-S6,F11	Event Report Send (ERS) VehicleDepositStarted . VehicleID = "AGVXX"
37.	Event Report Acknowledge (ERA)	S6,F12->		
38.			<-S6,F11	Event Report Send (ERS) CarrierRemoved . VehicleID = "AGVXX" . CarrierID = "654321" . CarrierLoc = "LOC2"
39.	Event Report Acknowledge (ERA)	S6,F12->		
40.			<-S6,F11	Event Report Send (ERS) VehicleDepositCompleted . VehicleID = "AGVXX"
41.	Event Report Acknowledge (ERA)	S6,F12->		
42.			<-S6,F11	Event Report Send (ERS) VehicleUnassigned . VehicleID = "AGVXX" . CommandID = "111111"
43.	Event Report Acknowledge (ERA)	S6,F12->		
44.			<-S6,F11	Event Report Send (ERS) TransferCompleted (Complete parameters) . CommandName = "TRANSFER" . CommandID = "111111" . Priority = 5 . Replace = 1 . ResultCode = 0 . $CarrierLoc_1$ = "PORTY1" . $TransferInfo_1$ – L,3 1. CarrierID = "123456" 2. SourcePort = "PORTX1" 3. DestPort = "PORTY1" . $CarrierLoc_2$ = "PORTZ1" . $TransferInfo_2$ – L,3 4. CarrierID = "654321" 5. SourcePort = "PORTY1" 6. DestPort = "PORTZ1"
45.	Event Report Acknowledge (ERA)	S6,F12->		

12.3 *Anomaly Transport*

12.3.1 *Host-Initiated CANCEL of a TRANSFER Command*

12.3.1.1 The Host wishes to terminate a previously issued TRANSFER command. This could be done so that the Host could issue another TRANSFER command with new command parameter values or so that a PGV could transfer the carriers. A Host initiated ABORT scenario would be similar to this scenario except that the remote command would be ABORT instead of CANCEL and the states when it is accepted by the TSC are different.

STEP	COMMENTS	HOST	TSC	COMMENTS
1.	The Host desires to CANCEL a particular TRANSFER command that it had previously issued to the TSC. Outstanding TRANSFER command has CommandId = "111111"			
2.	Host Command Send (HCS) **CANCEL** . COMMANDID = "111111" The COMMANDID must match that of the TRANSFER command that is being CANCEL'ed	S2,F41->		
3.			<-S2,F42	Host Command Acknowledge (HCA) This is the point where the TSC would reject the CANCEL command if not in the QUEUED or WAITING state.
4.			<-S6,F11	Event Report Send (ERS) TransferCancelInitiated . CommandID = "111111"
5.	Event Report Acknowledge (ERA)	S6,F12->		
6.			<-S6,F11	Event Report Send (ERS) TransferCancelCompleted . CommandID = "111111"
7.	Event Report Acknowledge (ERA)	S6,F12->		
8.			<-S6,F11	Event Report Send (ERS) VehicleUnassigned . VehicleID = "CARXX" . CommandID = "111111"
9.	Event Report Acknowledge (ERA)	S6,F12->		

STEP	COMMENTS	HOST	TSC	COMMENTS
10.	The host may now initiate another TRANSFER command for the carrier(s) from the TRANSFER command that was canceled or allow a PGV to deliver the carriers. If there were multiple carriers in the CANCEL'ed TRANSFER then the Host may elect to issue one TRANSFER command for all carriers or individual TRANSFER commands for each carrier. The SOURCEPORT in the resulting TRANSFER command(s) will be the same as when the original TRANSFER was issued since the carriers have not moved.			

12.3.2 *Host-Initiated Override of a TRANSFER Command*

12.3.2.1 The Host wishes to override the destination of a previously issued TRANSFER command after the command has entered the ACTIVE state. This could be done so that the Host could issue another TRANSFER command with a new destination.

STEP	COMMENTS	HOST	TSC	COMMENTS
1.	The Host desires to ABORT a particular TRANSFER command that it had previously issued to the TSC. The outstanding TRANSFER command has CommandID = "111111"			
2.	Host Command Send(HCS) **ABORT** . COMMANDID = "111111" The COMMANDID must match that of the TRANSFER command that is being ABORT'ed	S2,F41->		
3.			<-S2,F42	Host Command Acknowledge (HCA) This is the point where the TSC would reject the ABORT command if it is not possible for the transport system to abort the ACTIVE TRANSFER command.
4.			<-S6,F11	Event Report Send (ERS) TransferAbortInitiated . CommandID = "111111"
5.	Event Report Acknowledge (ERA)	S6,F12->		

STEP	COMMENTS	HOST	TSC	COMMENTS
6.			<-S6,F11	Event Report Send (ERS) TransferAbort-Completed . CommandID = "111111" . CarrierLoc = "LOC1" . TransferInfo
7.	Event Report Acknowledge (ERA)	S6,F12->		
8.			<-S6,F11	Event Report Send (ERS) VehicleUnassigned . VehicleID = "CARXX" . CommandID = "111111"
9.	Event Report Acknowledge (ERA)	S6,F12->		
10.	The host will now initiate another TRANSFER command for the carrier(s) from the TRANSFER command that was aborted.			
11.	Enhanced Remote Command(ERC) **TRANSFER** . COMMANDID = "111112" . PRIORITY = 5 . REPLACE = 0 . $TRANSFERINFO_1$ - L,3 1. CARRIERID = "123456" 2. SOURCEPORT = The current location of the carrier as sent from the TSC to the Host in the TransferCompleted for the failed TRANSFER command(an example is a port on a vehicle). The Host does not have to know the difference of whether the carrier is still on the car or not to send another TRANSFER command. 3. DESTPORT = "PORTYY"	S2,F49->		
12.			<-S2,F50	Enhanced Remote Command Acknowledge (ERCA)
13.	Scenario will now follow 12.1.1			

12.3.3 *Unsuccessful Completion of a TRANSFER Command*

12.3.3.1 The TSC must unsuccessfully complete a previously issued TRANSFER command due to an unrecoverable transport system error. Although this scenario follows the GEM defined scenario for Alarm handling, it was included to define the specifics of an IBSEM alarm and subsequent unsuccessful TRANSFER command completion. The abort of TRANSFER command should be performed by host as described in ¶12.2.2.

STEP	COMMENTS	HOST	TSC	COMMENTS
1.			<-S5,F1	Alarm Report Send (ARS) Alarm Set Unrecoverable Transport System Error . ALID . ALTX

STEP	COMMENTS	HOST	TSC	COMMENTS
2.	Alarm Report Acknowledge (ARA)	S5,F2->		
3.			<-S6,F11	Event Report Send (ERS) TransferPaused . CommandID
4.	Event Report Acknowledge (ERA)	S6,F12->		
5.			<-S6,F11	Event Report Send (ERS) AlarmSet . All VIDs necessary to fully describe the nature of the unrecoverable error must be sent to the Host in this event.
6.	Event Report Acknowledge (ERA)	S6,F12->		
7.	Human intervention is necessary to clear the unrecoverable error.			
8.	Note that the ResultCode is crucial for the Host to determine that the TRANSFER command has completed unsuccessfully. It is equally important that the current locations of all carriers is sent to the Host in the unsuccessful completion event.		<-S6,F11	Event Report Send (ERS) TransferCompleted . CommandName = "TRANSFER" . CommandID = "111111" . Priority . Replace . ResultCode = Error /*NonZero*/ . CarrierLoc = "LOC1" . TransferInfo
9.	Event Report Acknowledge (ERA)	S6,F12->		
10.			<-S5,F1	Alarm Report Send (ARS) Alarm Cleared Unrecoverable Transport System Error . ALID . ALTX
11.	Alarm Report Acknowledge (ARA)	S5,F2->		
12.			<-S6,F11	Event Report Send (ERS) AlarmCleared . All VIDs necessary to fully describe the nature of the unrecoverable error must be sent to the Host in this event.
13.	Event Report Acknowledge (ERA)	S6,F12->		

STEP	COMMENTS	HOST	TSC	COMMENTS
14.	The host may now initiate another TRANSFER command for the carrier(s) from the TRANSFER command that was completed unsuccessfully. If there were multiple carriers in the unsuccessful TRANSFER then the Host may elect to issue one TRANSFER command for all carriers or individual TRANSFER commands for each carrier. The SOURCEPORT in the resulting TRANSFER command(s) may be an internal vehicle location or any other location along the path of the TRANSFER command.			

12.3.4 *Connection or Reconnection between TSC and Host*

12.3.4.1 The Factory Host System crashes (or loses communication with the TSC for a time exceeding all time-outs and retries) and must re-synchronize with the TSC in lieu of several events completing while the communication link between the two was down.

STEP	COMMENTS	HOST	TSC	COMMENTS
1.	Communication session between host and TSC (re)established. Host establishes communication with the TSC per the GEM standard scenario (e.g., S1F13, etc).			
2.	Host Command Send (HCS) **PAUSE**	S2,F41->		
3.			<-S2,F42	Host Command Acknowledge (HCA)
4.			<-S6,F11	Event Report Send (ERS) TSCPauseInitiated
5.	Event Report Acknowledge (ERA)	S6,F12->		
6.			<-S6,F11	Event Report Send (ERS) TSCPauseCompleted . CommandName = "PAUSE" . ResultCode = 0
7.		S6,F12->		
8.	Selected Equipment Status Request (SSR) . ActiveCarriers . ActiveVehicles . ActiveTransfers	S1,F3->		HOST asks for carrier and vehicle information

STEP	COMMENTS	HOST	TSC	COMMENTS
9.			<-S1,F4	Selected Equipment Status Data (SSD) . CarrierInfo - L,3 . . (one CarrierInfo for each carrier) . . VehicleInfo - L,2 . . (one VehicleInfo for each vehicle) . . TransferCommand - L,n . . (one TransferCommand for each . Active TRANSFER command)
10.	The HOST updates its model of the system with the information from the vehicle and carrier status data.			
11.	Host Command Send (HCS) **RESUME**	S2,F41->		HOST enables the system to continue operations.
12.			<-S2,F42	Host Command Acknowledge (HCA)
13.			<-S6,F11	Event Report Send (ERS) TSCAutoCompleted
14.	Event Report Acknowledge (ERA)	S6,F12->		
15.				System continues processing all commands that were in process or queued before/during the Host crash or communication loss/initialization. System will also now process new commands.

12.4 *Database Operation*

STEP	COMMENTS	HOST	TSC	COMMENTS
1.	The host desires to upate the TSC database with information associated with carrier 123			
2.	Host Command Send (HCS) INFOUPDATE . CARRIERID = “123” . LOTID = “LOT456” . OPERATION = “OP480”	S2,F41->		The Lot and Operation information associated with the carrier are sent to the TSC.
3.			<-S2,F42	Host Command Acknowledge (HCA)

STEP	COMMENTS	HOST	TSC	COMMENTS
1.	The host desires to remove the LotID associated with carrier 123 from the TSC database			

STEP	COMMENTS	HOST	TSC	COMMENTS
2.	Host Command Send (HCS) INFOUPDATE . CARRIERID = "123" . LOTID = ""	S2,F41->		
3.			<-S2,F42	Host Command Acknowledge (HCA)

12.5 *Stage Delete Command*

STEP	COMMENTS	HOST	TSC	COMMENTS
1.	The host desires to delete ALL Stage commands.			
2.	Host Command Send (HCS) STAGEDELETE	S2,F41->		All stage commands are deleted. No parameters are included in the message.
3.			<-S2,F42	Host Command Acknowledge (HCA)

13 GEM Capabilities

13.1 The purpose of this section is to specify any SEMI E30 additional capabilities that are required to be supported by this class of equipment.

13.2 *Requirement*

13.2.1 This standard requires that the SEMI E30 fundamental requirements and additional capabilities have been implemented on the IBSEM equipment with the exception of Trace Data Collection, Remote Control, Process Program Management, and Limits Monitoring and Spooling. The TRANSFER Command State Model will serve as the Equipment Processing State Model specified in SEMI E30. If these capabilities are implemented, they will be implemented as required by SEMI E30. The SEMI E30 additional capabilities required by IBSEM are:

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

RELATED INFORMATION 1
IBSEM UNIQUE CAPABILITIES

NOTICE: This related information is not an official part of SEMI E82, but was approved for publication by full letter ballot procedures.

R1-1 Transfer Command Message Examples (SML Format)

R1-1.1 Variable data values specified in the following TRANSFER commands have been chosen arbitrarily for the purpose of demonstrating message structure/content.

R1-1.2 *Transfer Command Message Example for a Single Carrier Transfer*

```
S2,F49
<L [4]
        <U2     0>                                          /* DATAID */
        <A[0]   ‘’>                                         /* OBJSPEC */
        <A[8]   ‘TRANSFER’>                                 /* RCMD */
        <L [2]
                <L [2]
                        <A[11]  ‘COMMANDINFO’>              /* CPNAME1 */
                        <L[3]
                                <L[2]
                                        <A[9]   ‘COMMANDID’>    /* CPNAME */
                                        <A[6]   ‘111111’>       /* CPVAL */
                                >
                                <L[2]
                                        <A[8]   ‘PRIORITY’>     /* CPNAME */
                                        <U2     5>              /* CPVAL */
                                >
                                <L[2]
                                        <A[7]   ‘REPLACE’>      /* CPNAME */
                                        <U2     0>              /* CPVAL */
                                >
                        >
                >
                <L [2]
                        <A[12]  ‘TRANSFERINFO’>             /* CPNAME2 */
                        <L[3]
                                <L[2]
                                        <A[9]   ‘CARRIERID’>    /* CPNAME */
                                        <A[6]   ‘123456’>       /* CPVAL */
```