M20 standard requires that the "M20" coordinate system is fixed on the wafer, and is not affected by how the wafer is loaded on equipment. Also, as stated in the SEMI M20 standard, an orientation of "0" is for a wafer loaded on equipment with the primary fiducial towards the operator or "down".

R1-2.2.1 Often, no other information is required for measurements to be made on a wafer. However, the "M20" coordinate system may not provide sufficient accuracy to locate measurement sites on patterned wafers. This is a result of any of a number of possible reasons, all of which can be described as "experimental errors". For instance, wafers are not perfectly round, fiducial dimensions may vary, and equipment (both steppers that place patterns on wafers and metrology equipment) capabilities to determine the location of the wafer center and fiducial vary.

R1-2.2.2 In order to deal with these errors, metrology equipment suppliers have developed various strategies, as discussed in the introduction section. Perhaps because the SEMI M20 standard's reference to "origins" of "other coordinate systems" is in the same sentence with "reference points", many suppliers adopted the location of one (or more) reference point(s) as the origin for their pattern-based coordinate system(s). Without a common reference site, multiple coordinate systems resulted. More nefarious strategies were developed to compensate for "stage errors" on equipment. These usually entail a series of reference points required to "zero in" on the measurement site, and this made comparison of locations determined on different equipment next to impossible.

R1-2.2.3 MSEM defines the "M20P" coordinate system to be one which is identical to the "M20" coordinate system, if there were no experimental errors. In other words, the location of the origin and axes of the "M20P" coordinate system are offset slightly from the origin and axes of the "M20" coordinate system because of experimental errors. If the location of reference points, called "alignment sites" in MSEM, is defined to be the "M20" coordinates where they are "expected" to be, and equipment is designed to be able to "find" the alignment sites, given the various possible experimental errors, the "found" and "expected" SEMI M20 coordinates can be used to determine the "M20-to-M20P" coordinate system transformation, as explained in appendix 1. Alignment site information is specified through the use of the MSEM data items named AlignList, the CPNAME CP-ALIGNLIST and, the table type named ALIGN-DEF-LIST.

R1-2.2.4 Most metrology equipment cannot distinguish whether patterned wafer site location errors are due to the wafer, the layout on the wafer, or the equipment's ability to locate the sites. However, information that is available through the use of patterned-wafer alignment sites can provide a means for identifying potential metrology equipment problems. For instance, assume that the only pattern-layout location error on a wafer is that due to the establishment of the location of the wafer center and fiducial. For many users and metrology systems, this is a good assumption. If this is the case, then the MSEM data item named XlateData can be used to track this error. Although the error may result from multiple sources, being able to track it on various metrology equipment will enable users to apply statistical process control techniques to identify the specific sources.

R1-2.2.5 Sites specified for measurements or additional alignments may be found by metrology equipment at locations which deviate from their expected locations through either pattern layout errors or metrology equipment "stage" errors. Again, in a controlled manufacturing process, these combined errors should be normally distributed, and non-normal deviations may indicate possible metrology equipment problems. These types of errors shall be reported through the use of the MSEM data item named Offset.

R1-2.2.6 A minimum of two alignment sites are necessary to establish an "M20P" coordinate system on a wafer. Additional sites are often used, as in the example given in appendix 1. The system supplier shall document the requirements for ALIGNLIST information, and detail how any "site-by-site" alignment sites (that is, those needed to obtain better location accuracy in the neighborhood of measurement site) are associated with the specific SITELIST measurement sites.

R1-2.2.7 Alignment site location information may be either provided to or by equipment. Some equipment have the capability to find and report the locations of "reference sites" without prior knowledge of their locations on the wafer. In this case, the equipment shall report the location of both alignment and measurement sites in "M20" coordinates. When alignment site information is provided to equipment to establish an "M20P" coordinate, measurement site locations shall be reported in the "M20P" coordinate system.

R1-2.2.8 MSEM encourages users to define "M20P" locations in reference to the user's device layout design within an ideal SEMI M20 wafer coordinate system. An alternative is to define "M20P" locations by "training" on a "benchmark" system.

R1-3 *Layout of Rectangular Elements on a Silicon Wafer* — Equipment shall be capable of routine, automated operation without needing wafer layout information (e.g., field or die maps). However, having the capability to provide wafer layout information to metrology equipment from the host can be desirable. In some cases, the amount of information required to define measurement site locations is less if a pattern-element-based format is used. MSEM defines a means to do this in this section, if desired, based on SEMI standard SEMI M21.

R1-3.1 SEMI standard SEMI M21, “Specification for assigning addresses to rectangular elements in a Cartesian array”, is limited (for MSEM purposes) by the fact that nothing is specified about how the rectangular elements are located on the wafer. The SEMI M21 standard details how to assign “addresses” to elements and how to find the “array center” element. In this section, MSEM defines how these elements are located on a wafer, using the data item named “M21Data”, and how to establish within-element coordinate systems.

R1-3.2 There are users who want or require much more layout information than is provided for by SEMI M21, such as within-element structure details or element attribute information. This additional layout information is beyond the scope of MSEM, since it is considered to be information which only is needed to aid operator-interactive use of metrology systems.

R1-3.3 *MSEM “M21” Layout* — The first pattern element layout issue to be addressed is that of determining which SEMI M21 elements are to be included in the MSEM “M21” layout. MSEM defines the “M21” layout to include all elements which either wholly or partially are within the circumference of the wafer. Thus the MSEM “M21” element layout does not correspond to the pattern layout exactly, since some “M21” elements may not contain patterns.

R1-3.4 The second layout issue is that of how best to specify the element locations on the wafer. The MSEM approach is to specify the “M20P” coordinate for the lower left corner of the minimum number of elements needed to define the layout, along with the element addresses. For a non-tiled layout, the location of a single element is sufficient to establish the “M21” layout. For tiled layouts, the location of one element in each row or column is required. Note that the location of the lower left corner of an element may be outside the circumference of the wafer.

R1-3.5 Layout definition is supported only for host-to-equipment communications. The user is responsible for ensuring that the element addresses provided to the equipment agree with the SEMI M21 specification. The equipment need not check this, other than to ensure that there are not conflicts within the provided layout, and shall report results with element addresses as provided by the user.

R1-3.6 M21 layouts are established within the “M20P” coordinate system, and need not require any additional alignment site data than is needed to establish the “M20P” coordinate system. However, as with “M20P”, additional alignment may be necessary because of errors in either the pattern layout or the equipment's ability to locate features. OFFSET shall be used to report the location corrections that result from any within-element alignments.

R1-3.7 MSEM provides a means for element-based coordinate systems, if required. This capability is provided by the option of specifying “M21” as the coordinate system for ALIGN-DEF-DATA and SITEDEFDATA items. For element-based site locations, MSEM requires that SEMI M21 element coordinate systems have x and y axes parallel to the respective M20-based coordinate system axes, with their origins at the lower left corner of each element.

R1-4 *An Example of How an M20P Coordinate System is Established on a Silicon Wafer*

R1-4.1 The following example is fairly basic. For this example, the equipment does M20P alignment via a repeated two-step process. The first step is done at a low resolution, the second at a high resolution, and the process is done at two positions on the wafer.

R1-4.2 The equipment documentation states that 4 alignment sites are required. These are defined to the equipment via the table type named ALIGN-DEF-LIST, as detailed below. The order of the sites in ALIGN-DEF-LIST is not important. The sites are then selected via the CPNAME item named CP-ALIGNLIST, which is included in the PP-SELECT command. The order of the sites listed in CP-ALIGNLIST is important, and is as specified in the equipment's documentation. The first site is the alignment site for the first low resolution site, the second item is for the first high resolution site, the third item is the second low resolution site, and the fourth is the second high resolution site.

Table R1-1 ALIGN-DEF-LIST

Align-Name	*Coordx*	*Coordy*	*Coordsys*	*Align- Attribute(n)*
Coarse1	-60000	-200	M20P	
Fine1	-60020	-205	M20P	
Coarse 2	60000	200	M20P	
Fine2	59980	195	M20P	

R1-4.3 ALIGN-DEF-LIST

L,4

1. <Coarse1>
2. <Fine1>
3. <Coarse 2>
4. <Fine2>

R1-4.4 Using this information, the equipment will go to the nominal M20 location for Coarse, then "find" where it actually is. The offset between the nominal M20 location and the actual M20 location is then used to "find" Fine1. The actual M20 location of Fine1 is saved. The process is then repeated for Coarse 2 and Fine2. The equipment can now determine the M20 to M20P offset from the nominal and actual coordinates.

First, a summary of the data:

xN1=-60020	yN1=-205	Nominal x and y data for the first fine site
xA1=-59800	yA1=-150	Actual x and y data for the first fine site
xN2= 59980	yN2= 195	Nominal x and y data for the second fine site
xA2= 60060	yA2= 175	Actual x and y data for the second fine site

R1-4.5 The equipment first calculates THETA (Θ), using, for example, the formula:

$$\Theta = \tan^{-1}\left[\frac{MA - MN}{1 + MAMN}\right]$$

where MA and MN are, respectively, the slopes of the lines connecting the nominal and actual fine sites, in M20 coordinates, calculated as follows:

$$MA = \left[\frac{yA_2 - yA_1}{xA_2 - xA_1}\right] \quad MN = \left[\frac{yN_2 - yN_1}{xN_2 - xN_1}\right]$$

R1-4.6 The equipment then calculates DELTAX and DELTAY, using, for example, the formulas:

$$DELTAX = \left[\frac{C\,sin\,(\Theta) + D\,cos\,(\Theta)}{(sin\,(\Theta))^2 + (cos\,(\Theta))^2}\right]$$

$$DELTAY = \left[\frac{C\,sin\,(\Theta) - D\,cos\,(\Theta)}{(sin\,(\Theta))^2 + (cos\,(\Theta))^2}\right]$$

Where C and D are the adjusted site 1 coordinates in a rotation-adjusted coordinate system calculated, for example, using the formulas:

C = yA1 - ((xN1 sin Θ) + ((yN1 cos Θ)

D = xA1 - ((xN1 cos Θ) - ((yN1 sin Θ)

R1-4.7 The equipment can also calculate a SCALEFACTOR term to indicate the relative difference between the length of the vector connecting the nominal alignment sites and the length of the vector connecting the actual alignment sites. This can be used, for example, to judge whether there is a problem with the alignment process, since the difference between these two vectors should be small.

$$SCALEFACTOR = \frac{VA}{VN}$$

where VA and VN are the length of the vectors connecting the actual and nominal alignment sites, calculated using the formulas:

$$VN = \sqrt{\left(yN_2 - yN_1\right)^2 + \left(xN_2 - xN_1\right)^2}$$

$$VA = \sqrt{\left(yA_2 - yA_1\right)^2 + \left(xA_2 - xA_1\right)^2}$$

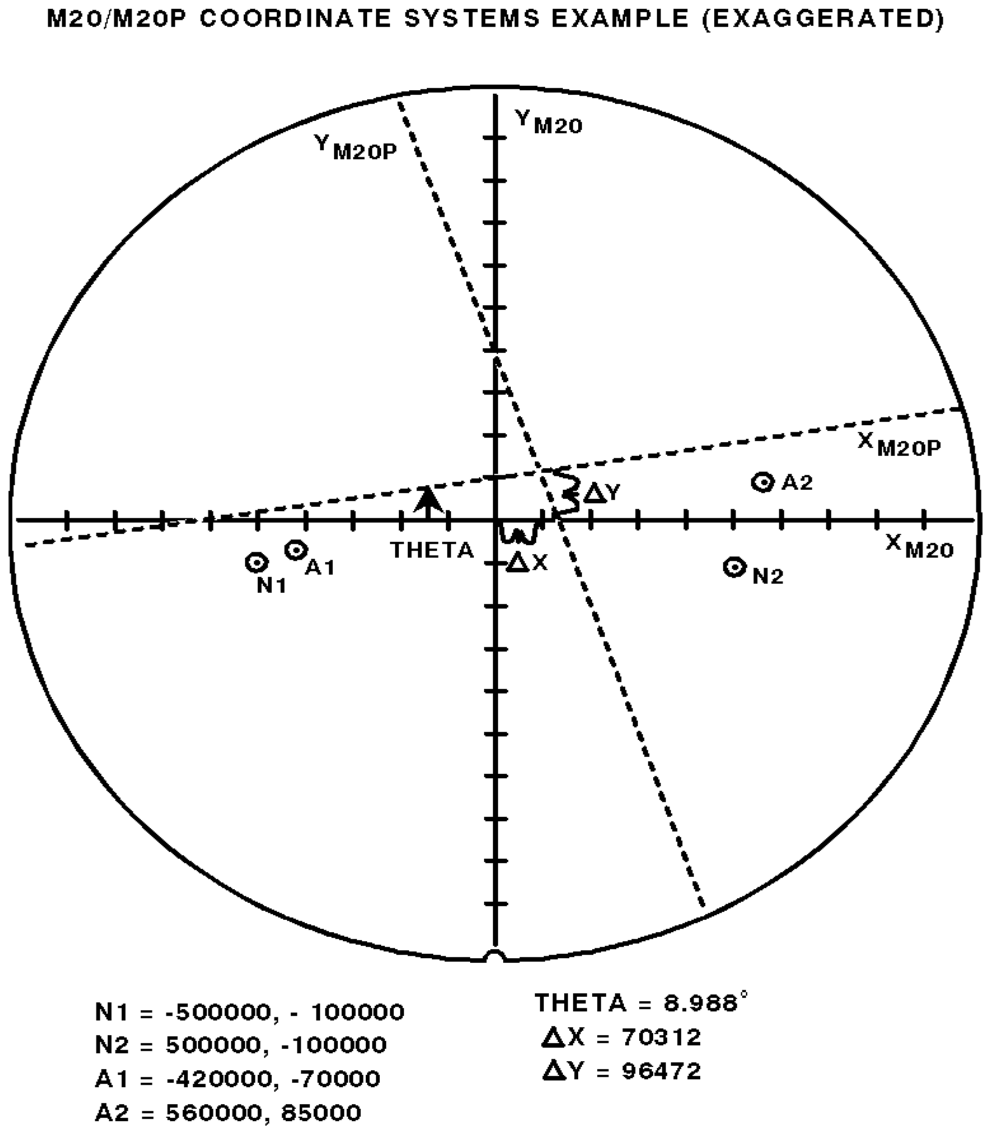

Figure R1-1
M20/M20P Coordinate Systems Example

R1-5 *Multi Measurement Station Metrology Equipment* — MSEM implementation on metrology equipment with multiple measurement stations should follow guidelines setablished for multi-processing station SEMS such as the Apply and Develop Track Specific Equipment Model (SEMATECH ID #: 95113021AENG Title: SEMATECH Apply/Develop Track Specific Equipment Model (ADTSEM), Version 0.8, January 4, 1996).

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

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

SEMI E32-0997
MATERIAL MOVEMENT MANAGEMENT (MMM)

CONTENTS

List of Figures

SEMI E32-0997
MATERIAL MOVEMENT MANAGEMENT (MMM)

1 Introduction

Automated material movement represents a significant milestone in the evolution of automation in semiconductor manufacturing. The standardization of the transfer process is critical to the economic viability of material movement automation. This standard addresses the communications needs of the semiconductor manufacturing facility with respect to material movement.

1.1 *Purpose* — This standard addresses automated material movement on the semiconductor factory floor—the task of transporting objects (material, et al) from one processing or storage location to another. It defines the concepts of material movement, the behavior of the equipment (including transfer devices) in relation to material movement, and the messaging services which are needed to accomplish the task.

1.2 *Scope* — The scope of this standard is defined from two viewpoints. The first is the breadth of the functionality covered. The second is the depth to which it is covered.

The breadth of functionality covered by this standard is limited to the activity required to transfer an object from a location on entity "A" to a location on entity "B" under the supervision of a factory host. This transfer may occur directly between two factory process equipment, or may involve the assistance of a "transfer agent", a device dedicated to material transfer. Thus, the material movement domain includes a maximum of four types of entities (see Figure 1.1):

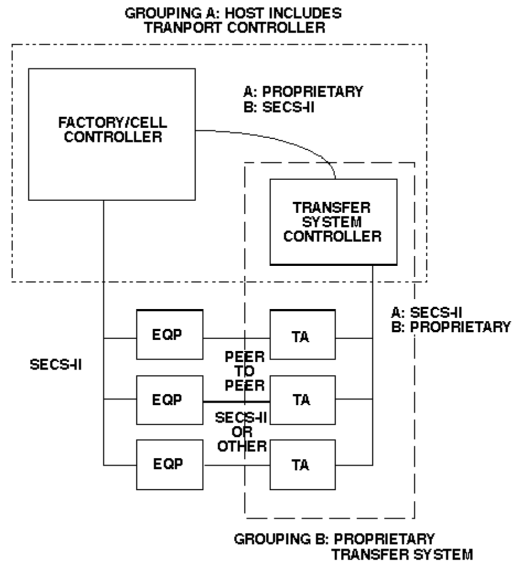

Figure 1.1
Illustration of Material Movement Scope

1. The factory <u>host</u>, which might, for example, be a factory controller or cell controller.
2. The set of process-related <u>equipment</u>, to which the host communicates.
3. The <u>transfer agent</u>, which performs the physical transport of the material in the factory. The transfer agent might communicate directly with the host or be controlled by an intermediate system.
4. This intermediate system, the <u>transfer system controller</u>, is the fourth type of entity.

This document assumes that the transfer system controller is either a part of the host or a part of a potentially complex transfer agent (as shown by the dotted boxes in Figure 1.1). Thus, the term "transfer system controller" seldom appears. The case where a "transfer agent" is actually a transfer system controller plus multiple transport devices is discussed further in Section 4.1.6.

There are some forms of material movement that do not fall within the domain of this standard and thus are not applicable to the methods described below. One example might be a non-deterministic system, such as a conveyer where generic parts are placed onto the system and circle until randomly selected by a workstation needing that type of part.

The subject of alarm reporting is not covered in this document. While material movement-related alarm situations will exist, the definition of the mechanism for reporting such alarms is left to other standards documents.

This standard presents a solution from the concepts and behavior down to the messaging services. It does not define the messaging protocol.

A messaging service includes the identification that a message shall be exchanged and definition of the data contained within that message. It does not include information on the structure of the message, how the data is represented within the message, or how the message is exchanged. This additional information is contained with the message protocol.

The defined services may be applied to multiple protocols. Information on the mapping of material movement services to specific protocols (e.g., SECS-II) are added as adjunct standards.

1.3 *References* — The following SEMI standards are related to the Material Movement standard:

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 SEMI Equipment (GEM)

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

Other References:

ISO/TR 8509:1987, Information Processing Systems, Open Systems Interconnection — Service Conventions

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

1.4 *Conventions*

1.4.1 *State Models*

Definition of behavior of systems or subsystems is based upon state models using the Harel notation. A discussion of this notation is available in the article by David Harel referenced above.

The Harel notation does not include the concept of "creation" and "deletion" of state models to represent transient entities. The "transfer job" described in this document is such an entity, where each new transfer job created uses a copy of the same state model. In this document, the use of an oval is used to denote the creation of an entity and also the deletion of that entity.

Transition tables are provided in conjunction with the state diagrams to describe explicitly the nature of each state transition. A transition contains columns for Transition #, Current State, Trigger, New State, Action(s), and Comment. The "trigger" (column 3) for the transition occurs while in the "current" state. The "actions" (column 5) include a combination of (1) actions taken upon exit of the current state, (2) actions taken upon entry of the new state, and (3) actions taken which are most closely associated with the transition. No differentiation is made.

1.4.2 *Object Services* — SEMI E39 Object Services Standard (OSS) requires that standardized objects be subtypes of the Top Object, that their attributes be defined in a table format and include specification of the value for object type, and that the objects support the GetAttr and SetAttr services. See SEMI E39 for details.

1 A brief tutorial on the Harel Statechart Notation is provided as Application Note A.5 of the SEMI E30 (GEM) standard.

Standardized objects are supported by an attribute definition table with the following column headings:

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

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

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

Form — Indicates the format of the attribute. (See Section 4.5.2 for definitions.)

2 Definitions

The following definitions are arranged in alphabetical order. Some are defined using terms defined elsewhere within this section. Please reference such definitions as needed. No references beyond this section should be necessary for a basic understanding of these terms.

Active Transfer Partner — (Opposite of Passive Transfer Partner) A transfer partner is considered active when it physically participates in the micro level portion of the transfer, either by moving the transfer object or by moving impediments within the transfer envelope (e.g., doors, clamps). This term refers to the micro level transfer phase only, and not to any setup activities prior to the transfer (e.g., a port door may be opened during setup phase by passive partner).

Atomic Transfer — The basic unit of movement. The transfer of a single transfer object from Equipment A directly to Equipment B where only one change in ownership occurs.

Compound Transfer — Combination of two or more atomic transfers executed sequentially or concurrently to achieve a single goal (e.g., exchange carriers or move a carrier between process machines using a transfer agent).

Dynamic Port — (Opposite of Static Port) A port with associated mechanisms capable of assisting with the physical movement of a transfer object or of interfering with the transfer of an object during the transfer. Such mechanisms may include doors, elevators, and robot arms. A transfer partner using a dynamic port for the transfer may be active or passive as required.

NOTE 1: The concepts of "dynamic" and "static" are associated with a port rather than equipment, as equipment may have both dynamic and static ports.

Equipment — Mechanical entity in the factory which plays a role in the manufacturing process. The equipment referenced in this document include machines used for processing, transport, and/or storage of material (see definition of material).

Host — In the context of material movement, the host is an entity, generally separate from either transfer partner, which coordinates and supervises a transfer job.

Interactive Transfer — A transfer in which both partners are active and must interact in the performance of the transfer.

Interfering — A dynamic port is interfering when any of its associated mechanisms are positioned where they are capable of physically obstructing the transfer.

Macro Level — Level of material movement that involves coordination by the host but may not require knowledge of the physical process used to accomplish the material transfer.

Material — A term used interchangeably with "transfer object" to refer to discrete objects which may be transferred to and from equipment. This may include product, carriers, reusable fixtures, etc. See transfer object definition.

Material Location — A physical position on a piece of equipment at which a transfer object may reside. Many material locations may be accessed directly through a port, but this is not a requirement. Some material locations internal to the equipment may not be accessible by a transfer agent.

Message Service/Service — A service (or a message service) represents a set of functions offered to a user by a provider. An unconfirmed service consists of a sequence of service primitives — the request from the sender to the communications facility and an indication to the receiver from the communications facility. Each of these service primitives is described by a list of parameters. A confirmed service adds a response to the initial request. The primitives for a response are called the response and the confirmation. A service excludes definition of message structure and protocol.

Micro Level — Level of material movement characterized by peer-to-peer interaction of the transfer partners to achieve synchronization of the detailed mechanical steps of material transfer.

Object — Webster's defines an object as “something perceptible.” In the software world, an object is a combination of attributes and behavior. It may refer to something concrete, such as a transfer object, or to a concept, such as a transfer job.

Ownership — An equipment is said to “own” a transfer object from the time the object is transferred into one of its ports until it is transferred out of the equipment. This indicates that the equipment has physical control of the transfer object.

Passive Transfer — A transfer that involves one active and one passive partner. During a passive transfer, the active partner retains control of the transfer envelope during the entire physical transfer.

Passive Transfer Partner — (Opposite of Active Transfer Partner) A transfer partner is considered passive when it takes no part in the physical micro level transfer, moving nothing within the transfer envelope. This term refers to the physical micro level transfer phase only. Setup activities prior to the transfer may be performed by a Passive Transfer Partner (e.g., a port door may be opened during setup phase).

Port — A point on the equipment at which a change of equipment ownership of a transfer object occurs. A port is not itself a material location, but shall have an associated location. A port may be thought of as an access point to an a material location on an equipment. The definition of the term port includes any dedicated mechanisms that either prepare for, facilitate, or are capable of interfering with the transfer. All equipment shall have a minimum of one port.

NOTE 2: While they are of use to the host system, the commonly used designations “Input Port” and “Output Port” are irrelevant in the context of this document. They are based on a flow of material through an equipment rather than the movement of transfer objects (which may be carriers rather than the material itself). Transfer objects may have to be moved into and out of both these types of ports.

Receiving Port — For a specific transfer, the port into which a transfer object is to be placed.

Sending Port — For a specific transfer, the port from which a transfer object is to be removed.

Static Port — (Opposite of Dynamic Port) A port with no associated mechanisms capable of assisting or interfering with the transfer of an object. A transfer partner utilizing a static port for the transfer shall always be passive.

Transfer Agent — An equipment specialized to the transport of material from one equipment (or storage area) to another.

Transfer Envelope — The three-dimensional space occupied during the transfer by the transfer object and all associated transfer mechanisms of both transfer partners. This defines the space in which transfer activity occurs and in which the potential for physical interference with the transfer exists.

Transfer Job — The set of atomic transfers constructed by the host to accomplish a cohesive material movement objective. See the section on compound transfers below.

Transfer Object — A physical object that is transferred to and from equipment, such as a product material, an empty carrier, or a carrier containing material to be processed. Tools (e.g., stepper reticles) and expendable materials also may be transfer objects. The term “material” is used interchangeably with “transfer object.”

Transfer Partners — In a given atomic transfer, the equipment sending a transfer object and the equipment receiving the transfer object are transfer partners.

Transfer Specification — The list of data provided by the host to define an atomic transfer.

Transfer System Controller — Entity that is responsible for management of multiple transfer agents. The transfer system controller presents a single communications interface to its host representing these multiple agents.

3 Overview

Material movement is concerned with the transfer of objects (WIP, tools, expendable materials, etc.) among process equipment, buffers, and storage facilities. These objects may optionally be contained (singly or in mass) within a carrier. Empty carriers may also be considered objects.

Material movement may include several different types of entities. These are divided into two classes: Host and Equipment. The host is the supervising entity which organizes the transfer and gives the appropriated instructions to each equipment involved. An equipment physically participates in the transfer, either sending or receiving material. The class of equipment may be divided into three types:

Process Equipment — Equipment which performs value-added operations on the factories’ products. This includes those equipment which physically changes the product and those which measure/test the product.

Transfer Agent — A type of equipment specialized to the movement of transfer objects from place to place within a factory. Transfer agents themselves may be of different types and with widely-differing characteristics. These may be fixed-arm robots, robot arms on a fixed track, AGVs with or without robot arms, overhead gantries, cars on tracks, or even transfer systems containing a heterogenous collection of other transfer agents. Humans also may act as transfer agents.

Storage Areas — A type of equipment used to hold transfer objects pending further processing (if product) or use (if reticles, etc.) within the factory. A storage area may be as sophisticated as an automated WIP Bin or as simple as a shelf.

3.1 *Macro and Micro Levels* — Material movement is separated into two levels. The more abstract level, called the macro level, consists of all message transactions used by the host in its coordination role. At the macro level, the host is responsible for determining what material is to be moved, from where, to where, and when. The host shall provide each party involved in a transfer job with the information necessary to perform its part of that transfer job.

A transfer job may be composed of multiple handoff operations, or "Atomic Transfers." Each atomic transfer involves only two equipment, termed "Transfer Partners." In order to perform this handoff operation, the transfer partners must achieve a level of synchronization. The communications between the transfer partners is called the micro level of material movement.

These communications occur between the two partners in a "peer-to-peer" fashion. Micro communications may be implemented via a direct equipment-to-equipment connection[2] or by passing messages indirectly through the host.

The macro and micro levels are designed to work together so that the host may initiate the transfer and then allow the transfer partners to perform the job interactively. However, these two levels may be used separately. For instance, the macro level may be combined with functionality of the SEMI E23 (Cassette Transfer Parallel I/O Interface Specification) standard. Additionally, under some conditions there may not be a host coordinating the transfer. Instead, one of the transfer partners might be in control. Here, the micro level may still be used to accomplish the transfer. The controlling partner would perform the coordination functions normally in the domain of the host.

In this standard, the macro and micro levels are defined separately, each with its concepts, behavior, and services.

2 In this context, connection is intended to mean a direct logical link, rather than a hard wired connection.

3.2 *Ports and Locations* — A port may be considered a point of entry to the equipment. A location is a receptacle for a transfer object. At any time, a port shall be linked to at least one location. However, some locations may be internal and never be linked to a port.

For "typical" factory equipment, there is a permanent one-to-one correspondence between the port and a location. However, not all semiconductor equipment follows this pattern. For instance, the port on a transfer agent that includes a robot arm and multiple on-board storage locations is actually the gripper on the robot arm. The gripper is also a location. In another case, a port may actually feed a rotating carousel containing multiple locations. The port in this example would be the access door to the carousel. The location linked to this type of port would be the one in front of the door at transfer time and would change whenever the carousel rotates. Figure 3.1 helps to illustrate the difference between ports and locations.

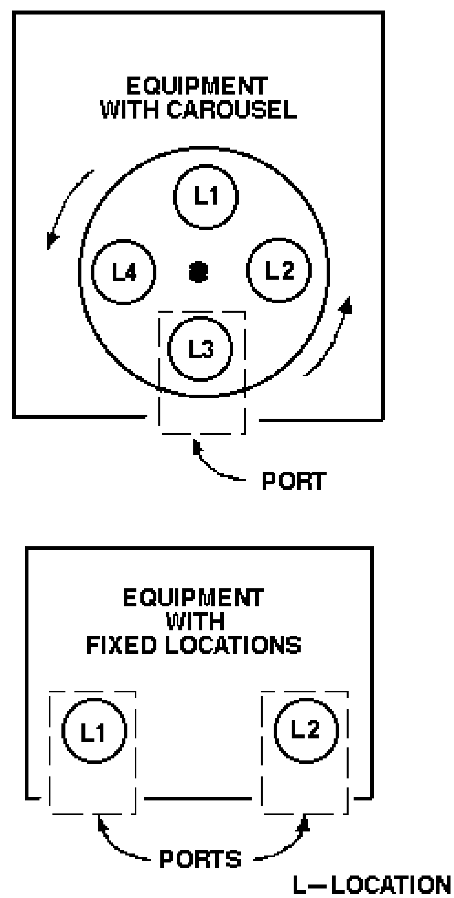

Figure 3.1
Port vs. Location

It is important to note that the attribute of “full” or “empty” belongs to a location rather than a port. A port that has just *received* a transfer object may be able to receive another immediately. Likewise, a port that has *sent* a transfer object may have more material available to send.

3.3 *Compliance* — Compliance to this standard includes adherance to all stated requirements in this document. This includes all defined messages services, state models, and communications scenarios.

This standard may be divided into two parts:

- Material Movement Macro Level, and
- Material Movement Micro Level.

An equipment may be compliant to one of these parts without the other. Details of compliance to each of these parts are defined in the body of this document.

4 Macro Level

The set of host<->equipment interactions needed to facilitate a transfer is called the macro level. These interactions concentrate on the definition of the transfer specification, and leave the details of the physical transfer to the micro level interactions.

In addition to the services defined in Section 4, the GetAttr service defined by SEMI E39, Object Services Standard, is supported by this standard. The definition for the GetAttribute service defined in Section 4 will be removed and no longer supported in 1998.

4.1 *Macro Level Concepts* — The macro level is designed with the intent of creating a widely applicable, host coordinated material movement capability. The result is a sequence of interactions from host to equipment that is used to accomplish the coordination of material movement required on the manufacturing floor.

The elements of material movement common to all transfers make up the framework. Other elements which differ from implementation–to–implementation are pushed, as much as possible, into the data portions of communications. The common elements identified are as follows:

- The host is responsible for designating the transfer partners which are compatible for transfer.
- The host must supply the necessary information for the transfer to each partner.
- Once transfer is defined, the responsibility moves to the transfer partners, especially the designated “primary” partner[3] . The host may monitor the transfer, but leaves the details to the transfer partners.
- The transfer partners inform the host of milestones during the process and of process completion.

Some of the elements which are more implementation-specific are:

- Which partner is the sender and which is the receiver of material. This has no intrinsic relationship to other details, such as the selection of the primary transfer partner.
- Whether a transfer partner is passive or actively participates in the transfer. This also becomes a data issue.
- The method (mechanism) of an interactive transfer. This may vary depending on the transfer partners' capabilities and system design. Interaction with different partners may require different methods. Transfer recipes containing this information may be specified by the host as a part of the transfer specification.

The host must have some knowledge of the capabilities of the transfer partners it designates to perform a transfer. It must understand which partners are compatible. If compatible, it must know the proper role for each partner of the transfer and what methods must be used. If an interactive transfer is warranted, the host must be able to determine which partner should be the primary. Definition of the mechanism for such determinations by the host is beyond the scope of this document.

4.1.1 *Macro Level Definitions*

Abort — The immediate termination of an active job, including a complete stop of all transfer job-related movement. An abort may be initiated by the host (and optionally by the operator). An abort by the host is directed at a transfer job, not at an individual atomic transfer. When told to abort, a transfer job shall, in turn, abort its atomic transfers. The abort command is intended to be used only in situations where there is risk of material or hardware damage.

Allocate — The formal reservation of an entity's resources for a specific purpose. In this document, it refers to the reservation of an equipment's resources for a specific transfer job or atomic transfer. Allocation of needed material locations for a transfer job occurs just prior to that job becoming active. Allocation of required port-related resources occurs just prior to the start of an atomic transfer.

Commit — The commit by the equipment indicates a readiness to begin the actual material handoff. The

3 See the definition for “primary transfer partner” in Section 4.1.1.

commit follows any setup activities and coincides with entry to the HANDOFF state. For example, an AGV may not be allowed to commit until it has moved to the point of transfer. The point at which an equipment may commit is equipment-specific. There is a separate commit for each atomic transfer in a job.

Deallocate — The release of allocated resources. In this document, it refers to the equipment's release of the resources reserved for a transfer job or an atomic transfer.

Port Resources — Equipment-controlled mechanisms that serve a port. An example of a port resource is a cassette indexer that changes to elevation of the material location. A port resource is available only to the atomic transfer that has allocated that port.

Primary Transfer Partner — The partner that controls the micro level transfer and that would receive the optional host command to initiate the transfer. The primary transfer partner shall always be an active partner for the transfer. For an interactive transfer, the nature of the transfer partners may determine which should be primary. Otherwise, the host may choose.

Restore — An operation associated with a transfer job which causes the resources used by that job to be returned to the preferred idle conditions (e.g., port access door closed). When a transfer job involves multiple ports on an equipment, restore activities are done on a port-by-port basis as each port is no longer needed by the transfer job.

Secondary Transfer Partner — The opposite of the primary transfer partner. This partner is either passive during the transfer or is controlled by the primary transfer partner. If active, this partner shall await communications from the primary partner before acting (see micro level).

Setup — A process associated with an atomic transfer that causes the port resources to achieve required pre–transfer conditions (e.g., port access door open).

Stop — A command available to the host that causes an orderly termination of a transfer job. Upon receipt of a stop command, the equipment shall complete all currently active atomic transfers, execute the "restore" process, and then terminate.

4.1.2 *Atomic Transfer* — An atomic transfer is the handoff of a transfer object from one equipment to another. This is the fundamental building block of material movement. An atomic transfer includes the minimum number of physical participants in the move: one sender and one receiver. Thus, only one change of ownership shall occur in an atomic transfer. All transfers required in practice can be constructed as a series of atomic transfers. A set of atomic transfers combined to make a complex but cohesive transfer is called a compound transfer.

4.1.3 *Compound Transfer* — On the factory floor, many material transfer situations will require multiple atomic transfers. For example, the host may determine that a material carrier needs to be moved from equipment "X" to equipment "Y" by transfer agent "T." To accomplish the transfer would require two atomic transfers, first "T" acquires the carrier from "X," then it delivers it to "Y." A compound transfer might involve several equipment and include parallel as well as sequential execution of atomic transfers.

4.1.4 *Parallel Transfers* — Each port on an equipment is a separate entity. It is possible, within the limitations of the equipment and transfer agents, to execute parallel transfers involving separate ports on an equipment. These parallel transfers may be with the same transfer partner, or with multiple partners. There are no inherent restrictions to a transfer job simultaneously executing multiple atomic transfers on an equipment. If an equipment's ports are static (i.e., always passive), the equipment should allow parallel transfers. If the ports are dynamic, there may be some sharing of resources which could limit this (e.g., use of the same robot arm). An equipment is not required to allow parallel transfers.

4.1.5 *Transfer Recipe* — The transfer recipe is an element of the transfer specification. The scope of a transfer recipe is one atomic transfer. A transfer recipe may contain information defining the following aspects of a transfer:

- setup/restore operations,
- sequence of handoff operations (see micro level),
- micro level commands to be issued by the primary partner to the secondary transfer partner,
- parameters relating to the transfer.

Thus, the host may use the transfer recipe to communicate the details that make it possible to prepare for and carry out the micro level transfer.

Use of transfer recipes is not a requirement. However, these recipes provide for dynamically "programming" an equipment to interact properly with what may be a completely different type of equipment. For instance, it may be possible to "teach" one transfer partner the proprietary commands that the other partner understands by embedding those commands in a transfer recipe. During the transfer, the primary partner could issue those commands at the proper time using standard micro level messages (see Section 5).

Transfer recipes should be created and managed according to applicable SEMI standards and in a similar manner to other types of recipes on the equipment (e.g., process recipes).

4.1.6 *Transfer System Controller as Transfer Agent* — A transfer agent is an entity to which transfer-related commands are given in order to carry out transfer jobs. A transfer system, which might include several material transport devices, can be described the same way. However, there are some important differences.

There are a number of ways that a host could view a transport system, including:

- as a single transfer agent with numerous ports,
- as a single transfer agent with a single port and numerous internal locations,
- as multiple logical transfer agents with their own ports, or
- as the set of real transfer agents that the transfer system controls.

The transfer system may conceal from the host the various internal operations (e.g., handoffs, travel paths) that it uses to complete a transfer. It may even hide the actual physical location (or series of locations) and provide a logical location for host reference.

4.1.7 *Transfer Job* — The host's material transfer objectives are defined in a transfer job. This transfer job may be a single atomic transfer or a compound transfer. A compound transfer job consists of multiple atomic transfers grouped together to accomplish a more complex objective. A typical compound transfer may also be accomplished by creating a separate transfer job for each required atomic transfer. The decision to group atomic transfers into a transfer job is application-dependent.

The key advantages to the use of compound transfer jobs are the reduction of the communication overhead to the host and the streamlining of the transfer process. The communication overhead is reduced by the fact that the host can define a number of atomic transfer operations with a single message to the equipment. The equipment are not required to wait for host commands between these atomic transfers.

The process is streamlined by virtue of the removal of redundant physical activities during the transfer job. Total setup and restore time for a port may be reduced if the setup for consecutive atomic transfers is similar.

One example of streamlining would be the “exchange” of carriers between equipment, where a transfer agent might bring an empty carrier to replace a full carrier on the “output port” of a machine. If the port has a door that is opened during transfer preparation and closed during the restore phase, the door would be better left open for both atomic transfers. Upon completion of the removal of the first transfer object, rather than close the door, the new transfer is begun and the door left open. The two atomic transfers would thus be executed as one smooth “swap” transfer without wasted time.

<u>Equipment View of Transfer Job</u> — A transfer job may involve a number of equipment and ports. Each of the involved equipment is given a Transfer Job Create request that specifies only the portion of the overall transfer job that involves that equipment. Only the host is guaranteed a complete picture of a compound transfer. In the example where a transfer job was created to move a transfer object from “X” to “Y” via transfer agent “T,”

- “X” would see <u>send material to “T,”</u>
- “Y” would see <u>receive material from “T,”</u> and
- “T” would see <u>get material from “X”</u> and <u>put material on “Y.”</u>

<u>Atomic Transfer Sequencing Guidelines</u> — When the transfer job given to an equipment contains multiple atomic transfers, the guidelines for performing those transfers are:

- Atomic transfers for a specified port are performed in the sequence given.
- When a transfer job references multiple ports on the equipment, these ports may execute their atomic job sequences in parallel to the other ports.

See Figure 4.1 for an illustration of the flow of a complex transfer job from the view of one equipment. The three parallel paths show a possible chronology for a transfer job.

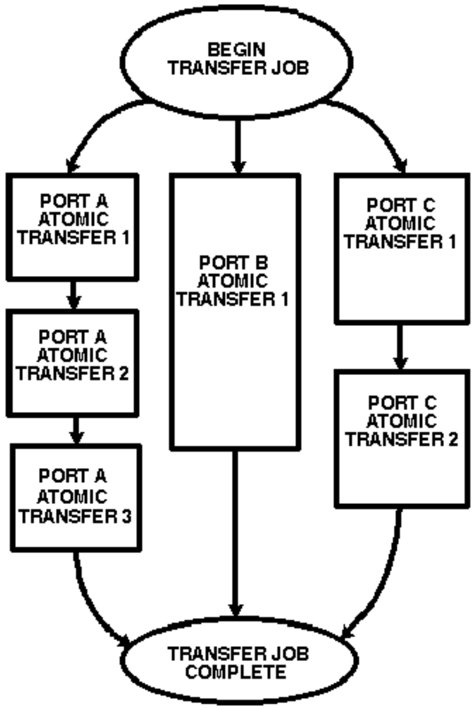

Figure 4.1
Transfer Job Sequencing

There are exceptions to the sequencing guidelines above. One case would be a device designed to simultaneously transfer from multiple ports on one machine to corresponding ports on another. This would translate into synchronized parallel atomic transfers on separate ports. Another case would be two atomic transfers involving the same port that might happen concurrently. That is, a transfer object is transferred out of the port at the same time[4] another is transferred in. In either of these cases, the equipment is responsible for:

1. recognizing that such transfers can and should be combined and
2. synchronizing the execution of the transfers.

Multiple Transfer Jobs — Multiple transfer jobs may exist simultaneously on an equipment. Although not a requirement, an equipment may

1. queue transfer jobs for later execution or
2. allow them to execute in parallel with other transfer jobs.

4 "At The Same Time" in this case means that both atomic transfers are in progress. It does not imply that both transfer objects must move simultaneously.

Queued transfer jobs and parallel transfer jobs are separate concepts, either of which may be supported.

Resource Allocation/Deallocation — Transfer jobs allocate the needed material locations prior to beginning. These material locations remain allocated until the transfer job chooses to release them. In many cases, a material location is physically linked with a single port, and the allocation of that location infers allocation of the port. However, this is not a requirement.

If the material locations for a transfer job are not available or are not in the proper state, the job shall be rejected or queued (if queuing supported and queue not full). For the transfer job to begin, each location referenced in the equipment's transfer job must contain the proper material for the first atomic transfer which would use that location.

If parallel transfer jobs are supported and the necessary locations are available, a second (or third, etc.) transfer job may begin execution.

In some cases a port may provide direct access to multiple locations. This may be true for the specialized ports described in Section 3.2 and in cases where the port serves as a pass–through to internal locations. For each atomic transfer, the needed port and its resources are allocated just prior to the start and released (deallocated) at the end of that atomic transfer.

4.2 *Macro Level Behavior*

4.2.1 *Macro Level Communications* — This section provides a high-level definition of the communications between the host and each transfer partner needed to achieve the macro level of material transport. This is not intended to define the messages, but rather to describe the concepts. The message detail is addressed in the Macro Level Services Section (4.4). Section 4.2.2 shows how these messages are integrated into the transfer job behavior model.

First, the control message flow is presented as Figure 4.2. Then additional informational messaging is described (see Figure 4.3). The arrows represent significant information exchange. Some replies that do not contain significant information may not be shown.

The ordering of the messages between one partner and host shall be retained. The key synchronization point between the partners at the macro level is that both partners shall complete their setup activities before the "atomic transfer started" event may occur.

4.2.1.1 *Macro Level Job Control* — Figure 4.2 illustrates the job control-related message flow expected on a normal transfer as seen by the host. These messages are used to control the material movement process. This diagram assumes a simple transfer job involving only two partners and only one port on each. It also assumes that the partners take the same role (primary or secondary) for each atomic transfer that occurs. The messaging is no different if these assumptions are not made, but diagrams become quite difficult to create.

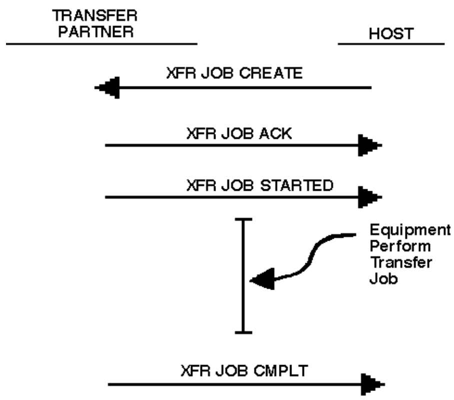

Figure 4.2
Macro Level Message Flow

Transfer Job Create — The host requests that the device participate in a specified transfer job. This may consist of a single atomic transfer or a compound job. This request may be acted upon immediately, or, if necessary resources are currently busy, saved for later execution[5]. The request shall supply a transfer specification for each atomic transfer, in which the host supplies such information as

- which port is to be used,
- whether the port is to send or receive material,
- identification of the material to be transferred,
- role of the equipment (primary or secondary transfer partner),
- mechanism or recipe to be used in the transfer,
- identification of the other transfer partner, and
- an identifier for the atomic transfer.

Upon receipt of the Transfer Job Create request and before acknowledging, the equipment should check the transfer specification(s) to ensure that they are valid. That is, that the specified parameters (ports, transfer recipes, role of the equipment, etc.) have legal values. Depending upon its ability to queue jobs, it may check such dynamic information as presence of transfer object, availability of resources, etc. to determine whether to accept or reject.

Transfer Job Create Acknowledge — The equipment responds to the host that the requested job is accepted or rejected, and if rejected, supplies the reason.

Transfer Job Complete — Once a transfer partner completes the restore operation, it declares the transfer job to be complete. This message is also used should a transfer job end abnormally. It declares the end of job. All movement of the involved mechanisms shall have ceased before this message is sent. This message shall provide information on the success or failure of the transfer.

Transfer Job Started — This message marks the beginning of the transfer job. It signifies that the setup activities for the first atomic transfer are beginning.

Get Attribute — The host may request information relating to a specific job. This may include portions of the transfer specification and current status information.

4.2.1.2 *Macro Level Information Messaging* — There are a number of material transfer-related events which may be of significance to the host. These are designated as “collection events” and shall be available for reporting to the host. This section describes those collection events and shows how they fit into the chronology of a transfer job. Refer to Figure 4.3 as each event is described.

Collection event messages provide valuable information to the host, but are not strictly required to perform material movement. Therefore, it is expected that some message protocol implementations will provide a method by which the factory host may disable those events which are not needed in a particular implementation.

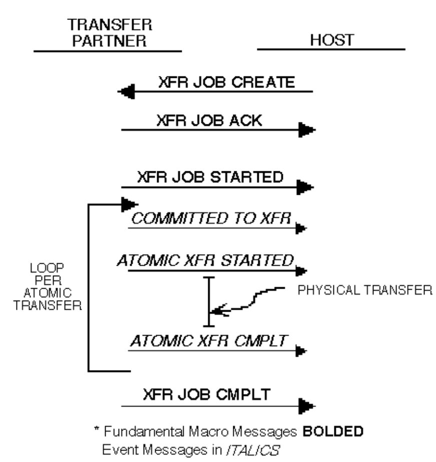

Figure 4.3
Macro Level Messaging With Events

5 If the equipment does not support queuing or if the queue is full, the request may be rejected.

Committed To Transfer — The equipment informs the host when it is fully prepared for a specific atomic transfer. This signifies the completion of the setup activities. It does not indicate that the two transfer partners have agreed to begin the physical handoff. This event coincides with the "Transfer Ready" message defined in the micro level (see Section 5 below).

Atomic Transfer Started — The transfer partners inform the host, via an event, when the atomic transfer has begun[6] . This may represent a logical start rather than physical movement. This indicates that any necessary coordination of the transfer partners has been done to ensure that the atomic transfer may start. A secondary partner may sense the start of transfer later than the primary partner.

NOTE 3: The physical transfer begins following the Atomic Transfer Started event and ends prior to the Atomic Transfer Complete event message.

Atomic Transfer Complete — This event indicates that an atomic transfer has completed. This indicates that either the next atomic transfer for this port will begin the setup phase, or, if none remain, restore activities may begin.

4.2.2 *Transfer Job State Model* — Message flow diagrams are useful to show simple situations. Material movement may range from very simple to quite complex. The transfer job state model presented in this section provides the information necessary to extrapolate the message flow diagrams above to fit all situations within the scope of this report. It does not cover the states of the various hardware which may exist (e.g., elevators, doors). See Section A1-1 for a discussion of the port hardware and related mechanisms.

The state model notation used in this report is that defined by Harel and is described in a reference cited in Section 1.3 above.

The transfer job is a transient entity. It is created by the host (Transfer Job Create message), executes, and then is dissolved by the equipment.

Figure 4.4 shows a basic state diagram for a transfer job. The ovals at top and bottom show where the job is created and then deleted. The portion between the ovals is standard Harel notation. This model is from an equipment view and assumes that only one port on the equipment is used by the transfer. It also assumes that pause, resume, abort, and stop activities do not exist (these are addressed later in the document).

6 The primary transfer partner will determine readiness to start based on micro level handshaking with the secondary transfer partner.

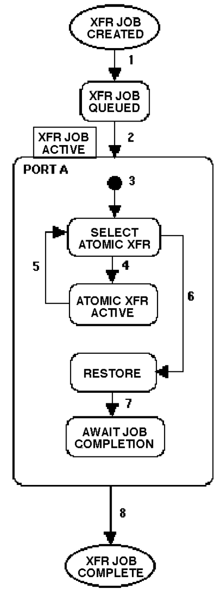

Figure 4.4
Transfer Job State Diagram for a Single Port

State definitions and then the state transition table follow.

TRANSFER JOB QUEUED

In this state, the transfer job has been accepted by the equipment through the Transfer Job Create/Acknowledge messages and is awaiting execution. Transfer jobs are begun sequentially in the order received by the equipment. Execution may not begin until all previously queued transfer jobs have been started and the material locations needed for this transfer job may be allocated.

All transfer jobs pass through this state. If the equipment allows queuing of transfer jobs, they may remain in this state for prolonged periods. A transfer job remains queued until the material locations needed for that transfer are available.

TRANSFER JOB ACTIVE

In the TRANSFER JOB ACTIVE state, all atomic transfers defined in the transfer job are performed.

PORT A

This substate of TRANSFER JOB ACTIVE encloses the job activities related to a single port on the equipment. Within this state, the atomic transfers involving this port are executed sequentially (except as noted in Section 4.1.7 under Atomic Transfer Sequencing Guidelines). Substates of PORT A (or PORT B, etc.) include SELECT ATOMIC

TRANSFER, ATOMIC TRANSFER ACTIVE, RESTORE, and AWAIT JOB COMPLETION.

SELECT ATOMIC TRANSFER

In this state, the next atomic transfer for this port is selected. Any prerequisites for the atomic transfer are checked (e.g., is material XYZ in Port A, Location 1). Atomic transfers are executed sequentially per port (except as noted in Section 4.1.7 under Atomic Transfer Sequencing Guidelines).

Prior to exiting this state, the port resources for the selected atomic transfer are allocated.

If no atomic transfer is ready for selection (e.g., the port resources are not available), the transfer job shall remain in this state, awaiting a change in conditions which would allow an atomic transfer to be selected.

ATOMIC TRANSFER ACTIVE

In this state, the next atomic transfer for this port is executed. See the atomic transfer state model for details on this process.

RESTORE

While in the RESTORE substate, the equipment shall perform any necessary post–transfer operations to return the port to the desired idle condition. These tend to be physical activities. No physical actions are required for restore, but some mechanism for determining port status is recommended.

The setup[7] and restore operations are expected to be "goal oriented," that is, oriented toward satisfying specific conditions (e.g., states such as "door open"). They should not depend on a fixed sequence of actions such as "initiate the door opening mechanism." Thus, a port left in a post–transfer condition from a previous atomic transfer would still be able to prepare properly for the next.

When queuing is allowed, the equipment may provide a look ahead capability so it may alter its restore phase to better accommodate the next transfer job. Such functionality is not required.

AWAIT JOB COMPLETION

In this substate, all atomic transfers on this port for the transfer job have completed. The transfer job continues to exist until all ports have completed their atomic transfers and restore operations. If only one port is used, this state will be occupied only briefly.

The following transition table defines the transitions between states for the transfer job state model. It applies to Figures 4.4 and 4.5.

7 Setup is more fully described in Section 4.2.3.

Table 4.1 Transfer Job State Transition Table

#	*Current State*	*Trigger*	*New State*	*Action(s)*	*Comment*
1	(Transfer Job Created)	The equipment accepts a Transfer Job from the host.	TRANSFER JOB QUEUED	Place the transfer job in the queue.	None.
2	TRANSFER JOB QUEUED	Necessary material locations have been allocated to transfer job and any equipment-specific prerequisites met.	TRANSFER JOB ACTIVE	Remove the transfer job from the queue. Send "Transfer Job Started" message.	None.
3	Any...	Default entry into the PORT A (or PORT B, etc.) state	SELECT ATOMIC TRANSFER	All atomic transfers are queued (by port).	None.
4	SELECT ATOMIC TRANSFER	Atomic transfer selected for execution on this port.	ATOMIC TRANSFER ACTIVE	Cause selected atomic transfer to transition from queued to setup.	See Sec. 4.2.3 for the atomic transfer state model.
5	ATOMIC TRANSFER ACTIVE	Atomic transfer complete event from the atomic transfer state model.	SELECT ATOMIC TRANSFER	None.	None.
6	SELECT ATOMIC TRANSFER	No atomic transfers remain for this port.	RESTORE	None.	None.
7	RESTORE	Port returned to desired idle conditions.	AWAIT JOB COMPLETION	None.	None.
8	TRANSFER JOB ACTIVE	All atomic transfers for the transfer job are completed.	(Transfer Job Complete)	Send "Transfer Job Complete" message.	None.

Figure 4.5 begins with the model in Figure 4.4 and is expanded to cover a transfer job with multiple ports. Each transfer job, as seen by the equipment, may include from one port to the total number of ports on the equipment.

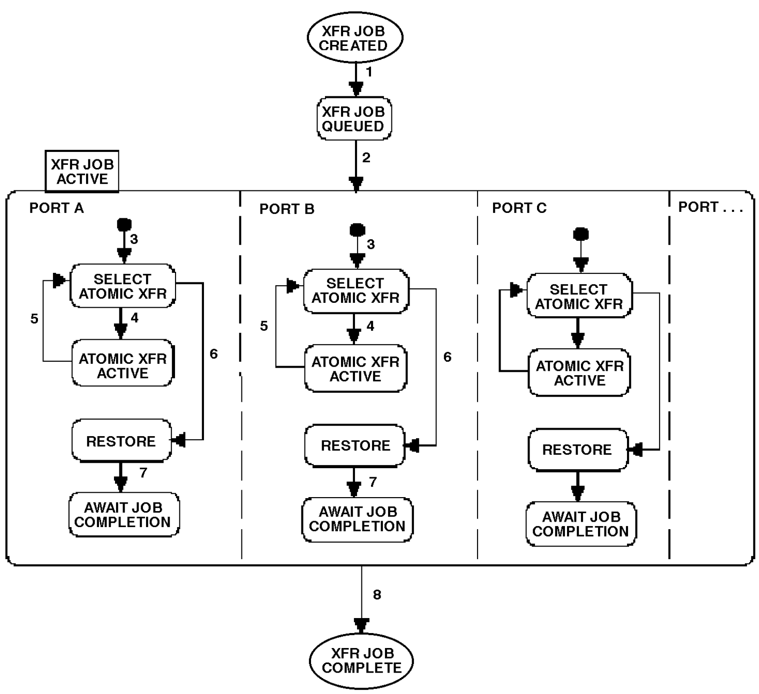

Figure 4.5
Transfer Job State Model for Multiple Ports

4.2.3 *Atomic Transfer State Model* — When a transfer job becomes active, it creates a queue of atomic transfers for each port (see Transition 2 of the transfer job state model). These atomic transfers are executed as described in the transfer job state model above. This section describes the behavior of the atomic transfer as it is created, executed, and finally deleted.

Figure 4.6 shows the state model for an atomic transfer as it moves through its life cycle. Each atomic transfer of an active transfer job is represented by a separate copy of this state model. These state models are not substates of the transfer job state model. Rather, transfer job and atomic transfers are separate entities which interact.

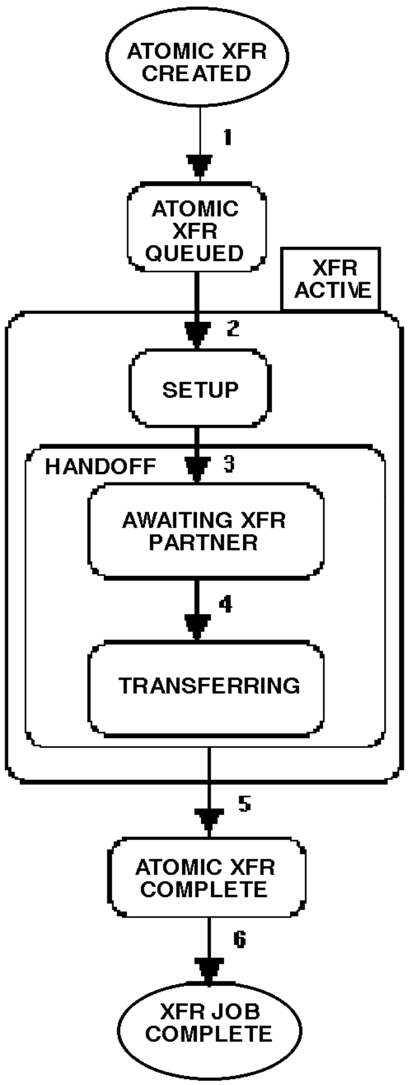

Figure 4.6
Atomic Transfer State Diagram

The definitions for the states and state transitions follow.

ATOMIC TRANSFER QUEUED

In this state, the atomic transfer is awaiting its turn to execute at the port.

TRANSFER ACTIVE

In this state, the transfer job has activated this atomic transfer. The atomic transfer is performed entirely within this state. The substates of TRANSFER ACTIVE are SETUP and HANDOFF.

The TRANSFER ACTIVE state in this model corresponds to the ATOMIC TRANSFER ACTIVE state in the transfer job state model. Thus, when the transfer job activates an atomic transfer on a port, both state models shall be in the (ATOMIC) TRANSFER ACTIVE state during the atomic transfer. This applies only to the specific atomic transfer being performed.

SETUP

While in the SETUP substate, the equipment performs any pre–transfer setup operations necessary for the current atomic transfer. These operations might include (but are not limited to) opening doors, unclamping carriers, and homing elevators. For transfer agents, it is reasonable to include as part of the setup such actions as a transfer agent moving into position near a specified process equipment.

The setup and restore[8] operations are expected to be "goal oriented," that is, oriented toward satisfying specific conditions (e.g., states such as "door open"). They should not depend on a fixed sequence of actions such as "initiate the door opening mechanism." Thus, a port left in a post–transfer condition from a previous atomic transfer would still be able to prepare properly for the next.

If no setup time is required for the transfer, this state may be exited immediately. This is the case for a static port since it can perform no physical actions related to setup.

HANDOFF

The equipment is committed to the transfer when no setup operations remain and the port is capable of performing the transfer. Entry into the HANDOFF state indicates that the port shall take no further action without the (expressed or implicit) cooperation of its transfer partner. Both transfer partners shall be in the HANDOFF state when the atomic transfer begins.

8 Restore is described in Section 4.2.2.

The HANDOFF substate is divided into two further substates: AWAITING TRANSFER PARTNER AND TRANSFERRING.

NOTE: If one transfer partner has not yet reached the HANDOFF substate when the other begins the transfer, interference could occur on the part of the uncommitted partner, with potential to damage equipment or material. The micro level Transfer Ready exchange will help prevent this (see Section 5 below).

AWAITING TRANSFER PARTNER

Once the port is ready for the transfer, the AWAITING TRANSFER PARTNER substate is entered. The port shall remain in this state until its transfer partner is ready.

TRANSFERRING

In this state, the physical handoff portion of the atomic transfer is in progress. Most micro level coordination occurs while the transfer partners are both in the TRANSFERRING state.

ATOMIC TRANSFER COMPLETE

In this state, the atomic transfer has completed and is awaiting the completion of the job. Upon entry to this state, the equipment shall release (deallocate) the port resources used for this atomic transfer. It shall also deallocate any material location used by this atomic transfer that is not needed by any further atomic transfer in this transfer job. This state exists primarily to allow the host to poll for current status of the individual atomic transfers (there is no status if the atomic transfer no longer exists).

The following transition table defines the transitions between states for the atomic transfer state model. It applies to Figure 4.6.

Table 4.2 Atomic Transfer State Transition Table

#	*Current State*	*Trigger*	*New State*	*Action(s)*	*Comment*
1	(Atomic Transfer Created)	The parent transfer job becomes active (see Table 4.1).	ATOMIC TRANSFER QUEUED	None.	None.
2	ATOMIC TRANSFER QUEUED	The parent transfer job selects this atomic transfer as next for a port and transitions to ATOMIC TRANSFER ACTIVE.	SETUP	None.	None.
3	SETUP	All preparation for this atomic transfer is complete.	AWAITING TRANSFER PARTNER	None.	"Committed To Transfer" event occurs. See micro level Transfer Ready message.
4	AWAITING TRANSFER PARTNER	Transfer partner is ready for transfer (see micro level).	TRANSFERRING	None.	"Atomic Transfer Started" event occurs. This transition might not occur for a static port.
5	HANDOFF	The atomic transfer is completed.	ATOMIC TRANSFER COMPLETE	None.	The "Atomic Transfer Complete" event occurs. This transition will occur after the appearance or removal of the material at the port. See the micro level Transfer Verify message.
6	ATOMIC TRANSFER COMPLETE	The parent transfer job is completed.	(Transfer Job complete)	The atomic transfer is deleted.	After this transition, the equipment maintains no status of the atomic transition.

4.3 *Extended Behavior Models* — In this section, the concepts of Pause/Resume, Stop/Cancel/Abort, and AutoStart/StartHandoff are added to the state models for the transfer job and the atomic transfer. In addition, the possibility that some transfers will end abnormally is discussed. While not every implementation requires these capabilities, all face the prospect of problems during transfer. This section describes how the basic state models provided above may be extended to include such features.

4.3.1 *Extended Communications* — This section describes the new commands the host may use to affect current transfer jobs and the events associated with these new capabilities.

Pause — The host may issue a command to pause a transfer job[9]. A command to pause shall cause the subject to continue to the first safe, continuable pausing place and then cease activity. The pause shall occur only at points that allow for the resumption of the activity (see the resume command). Pausing a transfer job results in the pausing of each atomic transfer for that transfer job (see Figure 4.8). The points where a pause may fall should be defined by the supplier. Note that a paused transfer job may be aborted or stopped as an alternative to the resume command. In this case, the stop command may cause the equivalent of a resume in order to allow the current atomic transfers to complete.

There are two actions that may cause a pause. The first is a host command as described in the paragraph above. The second is a decision by the equipment that a problem exists that requires outside intervention. For instance, if a port hardware problem exists, the equipment may pause the job automatically. The user or host may then resume the transfer job when the hardware problem is corrected, or alternately abort/stop the transfer job if the problem is severe.

Resume — The resume command is used to continue a previously paused transfer activity.

Stop — The host may command the involved partners to stop a transfer job. The stop command terminates a job in an orderly manner. The object of the stop command is to complete currently executing atomic transfers and leave the equipment and material in a safe state. While the time needed to stop should be kept as short as possible, this goal should be a lower priority.

If the specified transfer job is queued, the stop command acts as a cancel command.

Cancel — The host may cancel a transfer job which has not yet become active (e.g., a job which is queued). This is useful when it is unacceptable to affect transfer which is in–progress. A canceled job is removed from the queue and discarded. No physical action may be associated with canceling a transfer job. If the specified transfer job is active, the equipment shall reject this command.

Abort — The host may command the involved partners to abort a transfer job. The goal of the abort command is to end the transfer activities (especially movement) as quickly as possible, while retaining equipment and material integrity. Aborting a transfer job shall cause the individual atomic transfers to also abort. As with stop, the abort command terminates the transfer job. Different from the stop command is that equipment executing an abort command shall require manual intervention before further material movement operations are performed. Abort is intended for use when serious problems are detected and further damage needs to be prevented.

The abort command takes precedence over the stop, pause, and cancel commands. If the specified transfer job is queued, the abort command acts as a cancel command.

Transfer Job Create — The Transfer Job Create is not new, but is extended to include the “AutoStart” option. There is a parameter added to each atomic transfer definition which allows selection of this option. If not selected, the atomic transfer shall enter a WAIT state after completing SETUP operations, but before informing its transfer partner that it is ready to begin. An event signals the host that the equipment is awaiting instructions.

This feature allows the host added flexibility in synchronizing material movement activities. The host may manually synchronize two transfer partners or it may wish to align a material movement activity with another equipment or factory-related activity.

StartHandoff — The StartHandoff command is issued by the host to cause a specific atomic transfer to transition from the WAIT state to the TRANSFER ACTIVE state.

4.3.2 *Extended Transfer Job State Model* — Figure 4.7 presents an extended transfer job state model. This model provides for the capability to pause/resume, stop, and abort. Except for the extensions, this model is the same as that in Figure 4.5. Therefore, only the extensions are discussed in this section.

9 It is expected that a host may also pause a port or other hardware. However, this document does not specify these hardware-related commands.

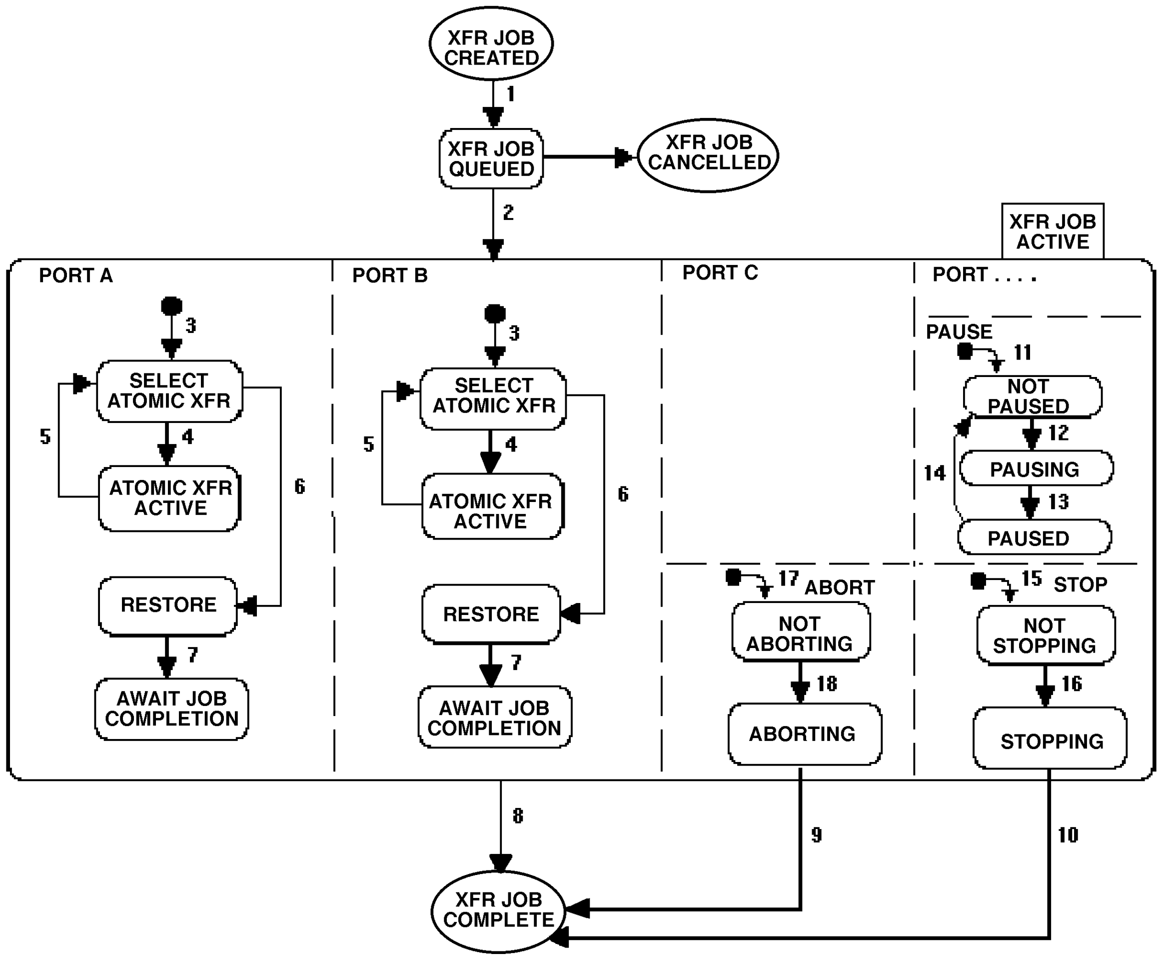

Figure 4.7
Extended Transfer Job State Model

ABORT

The ABORT state is an AND substate of TRANSFER JOB ACTIVE. This state is responsible for coordination of any activities necessary to abort[10] the active transfer job. The ABORT state has two substates: NOT ABORTING and ABORTING.

NOT ABORTING

In this state, the abort process is not in effect. The transfer job proceeds normally.

ABORTING

In this state the transfer job is causing its active atomic transfers to abort. This state is active only so long as any of the transfer job's atomic transfers remains active.

STOP

The STOP state is an AND substate of TRANSFER JOB ACTIVE. This state is responsible for coordination of any activities necessary to stop[11] the active transfer job. STOP has two substates: NOT STOPPING and STOPPING.

NOT STOPPING

In this state, the STOP process is not in effect. The transfer job proceeds normally.

STOPPING

In this state the process of stopping occurs. When the STOPPING state is active and the SELECT ATOMIC TRANSFER state is entered, that atomic transfer shall transition to the AWAIT JOB COMPLETION state. When all ports have reached that state, transition 10 from the STOPPING state occurs (see Figure 4.7).

10 See Section 4.1 for the defintion of abort.

11 See Section 4.1 for the definition of stop.

PAUSE

The PAUSE state is an AND substate of TRANSFER JOB ACTIVE. That is, the pause feature is concurrent with the execution of the transfer job. The PAUSE state has three substates: NOT PAUSED, PAUSING, and PAUSED.

NOT PAUSED

A newly activated job is NOT PAUSED by default. In this state, the transfer job proceeds as normal.

PAUSING

This state is coordinating the movement of the transfer job from NOT PAUSED to PAUSED. It is responsible for making sure that the current activity for each port for the job has ceased at an appropriate point. The transfer job is PAUSING until all of its active atomic transfers are paused.

PAUSED

All activity has ceased. The transfer job is awaiting a Resume command. Note that the PORT A, PORT B, etc., substates retain their current states, but their activity ceases when PAUSED.

The following transition table defines the added transitions between states for the extended transfer job state model. It applies to Figure 4.7.

Table 4.3 Extended Transfer Job State Transition Table

#	*Current State*	*Trigger*	*New State*	*Action(s)*	*Comment*
9	ABORTING	Abort process is complete.	(Transfer Job Complete)	Send “Transfer Job Complete” message to host indicating abnormal termination.	Host or equipment previously issued abort.
10	STOPPING	Stop process is complete.	(Transfer Job Complete)	Send “Transfer Job Complete” message to host indicating abnormal termination.	Host or equipment previously issued abort.
11	(Undefined)	Default entry into PAUSE state.	NOT PAUSED	None.	None.
12	NOT PAUSED	Host or equipment issued a directive to pause.	PAUSING	None.	None.
13	PAUSING	All transfer job activity has ceased.	PAUSED	None.	“Transfer Job Paused” event has occurred.
14	PAUSED	Host issued a directive to resume the transfer job.	NOT PAUSED	None.	“Transfer Job Resumed” event has occurred.
15	(Undefined)	Default entry into STOP state.	NOT STOPPING	None.	None.
16	NOT STOPPING	Stop initiated.	STOPPING	Begin stop activities.	None.
17	(Undefined)	Default entry into ABORT state.	NOT ABORTING	None.	None.
18	NOT ABORTING	Abort initiated.	ABORTING	Begin abort activities.	None.
19	TRANSFER JOB QUEUED	Cancel, Abort, or Stop command received.	TRANSFER JOB CANCELLED	Remove the transfer job from the queue.	None.

4.3.3 *Extended Atomic Transfer State Model* — Figure 4.8 presents an extended atomic transfer state model. This model extends the atomic transfer model in a way similar to the transfer job extension. It provides for the capability to pause/resume and to abort[12]. Except for the extensions, this model is the same as that presented as Figure 4.6. Therefore, only the extensions are discussed in this section.

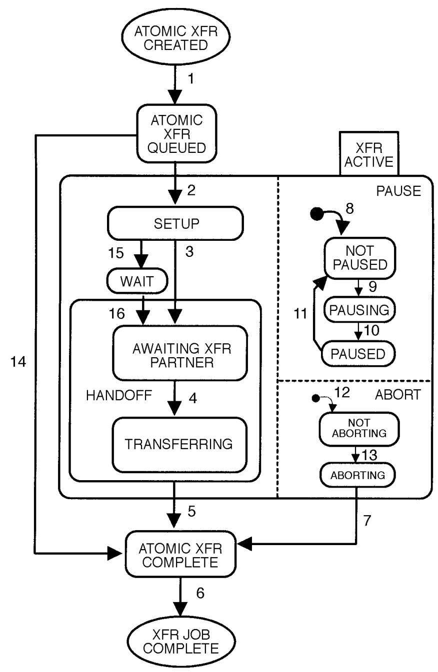

Figure 4.8
Extended Atomic Transfer State Diagram

WAIT

The atomic transfer stops activity and awaits a StartHandoff command from the host. This capability allows the host to precisely orchestrate the timing of the handoff in time critical operations.

ABORT

The ABORT state is an AND substate of TRANSFER ACTIVE. This state is responsible for coordination of any activities necessary to abort the active atomic transfer. The ABORT state has two substates: ABORTING and NOT ABORTING.

NOT ABORTING

In this state, the abort process is not in effect. The atomic transfer proceeds normally.

ABORTING

In this state, the process of aborting is performed. It is the responsibility of the equipment to cease physical activity as quickly as possible and end the current atomic transfer. This process is application-specific. It is the responsibility of the supplier to document this method.

PAUSE

The PAUSE state is an AND substate of ATOMIC TRANSFER ACTIVE. That is, the pause states are concurrent with the execution states of the atomic transfer. The PAUSE state has three substates: NOT PAUSED, PAUSING, and PAUSED.

NOT PAUSED

A newly activated atomic transfer is NOT PAUSED by default. In this state, the atomic transfer proceeds as normal.

PAUSING

This state is coordinating the movement of the atomic transfer from NOT PAUSED to PAUSED. It is responsible for making sure that the current activity for the atomic transfer has ceased at an appropriate point.

PAUSED

All activity has ceased. The atomic transfer is awaiting a Resume command. Note that the PORT A, PORT B, etc., substates retain their current states, but their activity ceases when PAUSED.

The following transition table defines the added transitions between states for the extended atomic transfer state model. It applies to Figure 4.8.

12 Note that “stop” does not apply to atomic transfers, since the active atomic transfers are allowed to complete when the transfer job is stopped.

Table 4.4 Extended Atomic Transfer State Transition Table

#	Current State	Trigger	New State	Action(s)	Comment
3	SETUP	Preparation for atomic transfer is complete, and TRAutoStart parameter is TRUE.	AWAITING XFR PARTNER	None.	"Committed to Transfer" event
7	ABORTING	Abort process is complete.	ATOMIC TRANSFER COMPLETE	None.	"Atomic Transfer Complete" event occurs noting abnormal termination.
8	(Undefined)	Default entry into PAUSE.	NOT PAUSED	None.	None.
9	NOT PAUSED	The transfer job transitioned to PAUSING state (Transition 12, Figure 4.7).	PAUSING	None.	None.
10	PAUSING	All atomic transfer activity has ceased.	PAUSED	None.	None.
11	PAUSED	The host issued a directive to resume the atomic transfer.	NOT PAUSED	None.	None.
12	(Undefined)	Default entry into ABORT.	NOT ABORTING	None.	None.
13	NOT ABORTING	Transfer job transitioned to ABORTING state (Transition 18, Figure 4.7).	ABORTING	Begin abort activities.	May be host- or equipment- initiated.
14	ATOMIC TRANSFER QUEUED	Abnormal Transfer Job termination.	ATOMIC TRANSFER COMPLETE	None.	May be caused by transfer job abort or stop.
15	SETUP	Preparation for atomic transfer is complete, and TRAutoStart parameter is FALSE.	WAIT	Wait for host command.	None.
16	WAIT	StartHandoff command is received from host.	AWAITING XFR PARTNER	None.	"Committed to Transfer 2" event

4.4 *Macro Level Services* — This section defines the messaging services required to implement the material movement concepts. The messages were introduced in Sections 4.2.1 and 4.3.1. These services are independent of the messaging protocol to be used. They may be mapped to SECS–II (SEMI E5) or to other comparable protocols.

These messaging services define the messages to be used, the nature of the data items or parameters to be contained within the messages, and data type of the parameters. Not defined here is the internal structure of the actual messages as transferred, including order of the parameters and how various data structures and data types are represented.

4.4.1 *Service List* — The following messages are exchanged between host and equipment for the purpose of accomplishing material movement tasks.

Host Initiated Services

Service Name	*Description*	*Confirmed**
TRJobCreate	Host request that a transfer job be performed.	Yes
TRJobCommand	Command which affects a transfer job.	Yes
GetAttribute	Request for attributes of an object (e.g., transfer job or atomic transfer). This message service has applicability beyond material movement.	Yes

* An unconfirmed service requires no response to the request message. This means that no data is required in the response. This does not preclude the implementation of a response message in the message protocol.

Equipment Initiated Services

Service Name	*Description*	*Confirmed*
TRJobAlert	Notification to host that transfer job is started or complete.	No
TrJobEvent	Notification to host that a transfer-related event has occurred.	No

4.4.2 *Service Detail* — The tables below define the parameters for each service. In some cases, parameters have additional detail which is defined in a following section. These parameters are marked with the "*" character.

The columns labelled REQ/IND and RSP/CONF link the parameters to the direction of the message. The message send by the initiator is called the "Request." When receiver terms this message the "Indication" or the request. The receiver may then send a "Response," which the original sender terms the "Confirmation."

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

"M" — Mandatory Parameter—must be given a valid value.

"C" — Conditional Parameter—may be defined in some circumstances and undefined in others. Whether a value is given may be completely optional or may depend on the value of other parameter.

"U" — User Defined Parameter.

"–" — Not Used.

"=" — (for Response only) Indicates that the value of this parameter in the response must match that in the primary (if defined).

The column labelled "Form" is used to indicate the type of data contained in a parameter. The forms used in this document are defined below.

Unsigned Integer — May take the value of any non–negative integer. Messaging protocol may impose a limit on the range of possible values.

Enumerated — The parameter may take on one of a limited set of possible values. These values are generally given logical names but may be represented by any data type excluding lists and structures.

Boolean — The parameter may take on one of two possible values, equating to "True" and "False."

Text — A text string.

Structure — A complex structure which consists of a collection of values of one of the possible forms. The breakdown of all "Structure" parameters is provided within this document.

NOTE 4: To prevent the definition of numberous variables named "XxxList", this document adopts the convention of referring to the list as "(List of) Xxx." In this case, the definition of the variable Xxx will be given, not of the list. The term "list" indicates an ordered collection (or set) of zero or more items of the same data type. List order is retained from the message request to the response. For this document, a list must contain at least one element unless zero elements are specifically allowed.

TRJobCreate

The host requests that the equipment participate in a material transfer job.

Parameter	*REQ/IND*	*RSP/CONF*	*Description*	*Form*
TRJobName	C	-	Host created identifier of the transfer job. This allows the host to use a common name for the transfer job on each participating equipment.	Text
(List of) AtomicSpec	M	-	Specifications for the atomic transfer(s).	Structure
TRJobID	-	M	Equipment created ID of the transfer job.	Unsigned Integer
(List of) TRAtomicID	-	M	The equipment-created identifier of the atomic transfer(s).* Order matches that given in the (List of) AtomicSpec in the message req.	Unsigned Integer
TRStatus	-	M	Reports acceptance or rejection of the transfer job.	Structure

*All object IDs created by the equipment must be unique within that equipment for that object type. This includes TRJobID and TRAtomicID. These IDs are used by the host to inquire about those objects.

TRStatus Parameter Detail

Parameter	*RSP/CONF*	*Description*	*Form*
TRAck	M	Tells whether the activity was successful (=true) or unsuccessful (=false).	Boolean
(List of) Status	C	Reports any errors found. (May be list of 0.)	Structure

Status Parameter Detail

Parameter	*RSP/CONF*	*Description*	*Form*
ErrorCode	M	Contains the code for the specific error found.	Enumerated
ErrorText	M	Text in support of the error code.	Text

AtomicSpec Parameter Detail

Many of these parameters are conditional. The equipment shall decide if it has a sufficient amount of information to perform each atomic transfer.

Parameter	*REQ/IND*	*Description*	*Form*
TRLink	M	Used by the transfer partners to assure that their communications relate to the same atomic transfer. The host is responsible for assuring uniqueness.	Unsigned integer
TRPort	C	Material I/O port on the equipment through which the transfer is to occur. If not specified, the equipment shall select the port.	Unsigned integer
TRObjName	C	Textual identifier of the object to be transferred.	Text
TRObjType*	C	Identifies the type of transfer object. Not required if only one type is acceptable.	Enumerated
TRRole	M	Equipment's role (i.e., Primary or Secondary).	Enumerated
TRRecipe	C	Name of the transfer recipe (if needed).	Text
TRPartner	M	Identifier (Name* attribute of equipment) of the transfer partner.	Text
TRPartnerPort	C	Material I/O port on the equipment's partner to be used for this atomic transfer.	Unsigned integer
TRDirection	M	Send or Receive the transfer object?	Enumerated
TRType	M	Is the equipment to be Passive or Active?	Enumerated
TRLocation	C	Identifier of the location on the equipment of the source (or destination) of the transfer object.	Unsigned integer
TRAutoStart	M	Should the equipment start the handoff without waiting for the “StartHandoff” command.	Boolean

TRJobCommand

The host requests a change in an active transfer job. Possible commands include Pause, Resume, Stop, Cancel, Abort, StartHandoff.

Parameter	*REQ/IND*	*RSP/CONF*	*Description*	*Form*
TRObjID	M	-	Equipment's identifier of the transfer job to which the command applies.	Unsigned integer
TRCmdName*	M	-	Indicates which command to perform.	Text
(List of) CmdParameter	C	-	Parameter(s) corresponding to the command type. (May be list of 0)	Structure
TRStatus	-	M	Describes the acceptance or rejection of the command.	Structure

CmdParameter Detail

Parameter	*RSP/CONF*	*Description*	*Form*
CmdParmName	M	The name of the parameter.	Text
CmdParmValue	M	Value of the parameter.	Varies per Parameter

TRJobAlert

The equipment indicates the start or completion of a transfer job.

Parameter	*REQ/IND*	*Description*	*Form*
TRJobID	M	Equipment's identifier of the transfer job.	Unsigned integer
TRJobName	M	Host created identifier of the transfer job.	Text
TRJobMS*	M	Transfer job milestone.	Enumerated
TRStatus	M	Status of the transfer milestone (i.e., success or failure).	Structure

GetAttribute

Information related to a specific object (e.g., transfer job) is requested.

Parameter	*REQ/IND*	*RSP/CONF*	*Description*	*Form*
ObjType*	M	-	Type of object for which attributes are desired.	Enumerated
(List of) ObjectID	C	-	Identifier of each object for which attributes are desired. If omitted (i.e., list of zero elements), it implies a request for all objects of the specified ObjType.	Varies according to ObjType.
(List of) AttributeID*	M	-	Identifier of requested attribute.	Enumerated
(List of) AttrData	-	C	Attribute data. Returns the requested attribute list for each specified ObjectID (e.g., grouped per ObjectID). If the object does not exist, an empty list is returned for that object, if an attribute does not exist, a null member of the list is returned. If the ObjType is unknown, a single null list is returned.	Varies per attribute.
(List of) Status	-	C	Reports any errors found in the GetAttribute request. (May be list of 0.)	Structure

TRJobEvent

The equipment reports pertinent state changes related to a transfer job.

Parameter	*REQ/IND*	*Description*	*Form*
TREventID*	M	Identifier of the specific event which occurred.	Enumerated
TRJobID	M	Equipment's identifier of the job to which the event refers.	Unsigned integer
TREventData	C	Data related to the specific event. May be repeated if multiple data parameters are reported.	Varies per parameter reported.

4.4.3 *Parameter/Attribute Definitions* — This section gives further details on parameters not sufficiently defined above. It also defines attributes of the defined object types. Note that some parameters also serve as attributes (e.g., TRJobName).

AtomicSpec — If a list of AtomicSpec parameters is supplied, the order is important. The equipment shall execute atomic transfers in the order specified on a per–port basis.

AttributeID/AttrData — An attribute is a characteristic which applies to all instances of that (type of) object. Thus, the attributes available for an object depend upon the ObjType. The three object types available for material movement are TransferJob and AtomicTransfer, and Equipment. Below are listed the defined attributes for each.

Object (i.e., any object): ObjectID, ObjType.

TransferJob: TRJobState, TRAtomicList, TRJobName, TRJobID (=ObjectID).

AtomicTransfer: TRAtomicState, TRLink, TRPort, TRObjName, TRRole, TRRecipe, TRPartner, TRPartnerPort, TRDirection, TRType, TRLocation, TRAutoStart, TRObjType, TRAtomicID (=Object ID).

Equipment: Name (=ObjectID).

ErrorCode — This parameter is used to report the reasons why an activity was unsuccessful. This applies to the TRJob Create, TRJobComplete, and the TRJobCommand. While it is impossible to define all possible errors, the following is a list of values which may apply in many cases:

TRJobCreate:

- Parameters improperly specified.
- Insufficient parameters specified.
- Queue full or queuing not allowed.
- Error in atomic transfer specification.
- Material Location Status does not match Transfer Specification.
- Primary Transfer Partner Role not supported.
- Active Participation not supported.

TRJobComplete:

- Failed due to hardware error.
- Failed due to transfer partner error.
- Failed due to error during atomic transfer.
- Transfer Job aborted by host.
- Transfer Job stopped by host.
- Transfer Job canceled by host.

TRJobCommand:

- Parameters missing or improperly specified.
- Requested service not available from this equipment.
- Cancel not allowed after transfer start.
- Unknown Transfer Job.

GetAttribute:

- Invalid object type.
- Unknown objectid.
- Invalid attribute.

ObjType — The type or class of the object of interest. The three types of objects contained in this document are TransferJob, AtomicTransfer, and Equipment.

Timestamp — The format for this Text string is YYYYMMDDhhmmsscc where:

YYYY= Year

MM= Month (01–12)

DD= Day (01–31)

hh= Hour (00–23)

mm= Minute (00–59)

ss= Second (00–59)

cc= Hundredths of Second (00–99)

TRCmdName/CmdParameter — Possible values are:

Command	*Parameter*
CANCEL	none
PAUSE	none
RESUME	none
ABORT	none
STOP	none
STARTHANDOFF	TRAtomicID

TREventID/TREventData — The following table shows the defined events and the data which are required to be available. Any attribute of the object should be available for reporting with an associated collection event.

Event	Typical Data
Committed To Transfer	TRAtomicID, Timestamp
Waiting for StartHandoff	TRAtomicID, Timestamp
Transfer Job Paused	Timestamp
Transfer Job Started	Timestamp
Atomic Transfer Started	TRAtomicID, Timestamp
Atomic Transfer Complete	TRAtomicID, Timestamp

TRJobMS — Identifies which transfer job milestone is being alerted to the host. Possible values are:

- Transfer Job Started.
- Transfer Job Complete.

TRObjType — Identifies the type of object to be transferred. This parameter is used only in cases where a port is capable of transfers involving different object types. As an example, a list of possible values might include:

- Wafer Reticle Cassette
- 100 mm Wafer Carrier
- 125 mm Wafer Carrier
- 150 mm Wafer Carrier
- 200 mm Wafer Carrier
- 100 mm Wafer
- 125 mm Wafer
- 150 mm Wafer
- 200 mm Wafer

Attributes: (Note: Some parameters above are also attributes.)

Name — Each equipment must have a unique Name attribute, a factory-defined Text string.

TRAtomicList — List of the atomic transfers (TRAtomicIDs) contained within the specified transfer job. This is an attribute of a transfer job and is specified in the GetAttribute service.

TRAtomicState — Attribute of an atomic transfer. Possible values are defined by the state models defined in Sections 4.2.3 and 4.3.3.

TRJobState — An attribute of transfer job. Possible values are defined by the state models defined in Sections 4.2.2 and 4.3.2.

4.5 *Object Attribute Definitions* — This section defines the attributes of those objects required for Material Movement Management: Equipment, TransferJob, and AtomicTransfer.

4.5.1 *Equipment* — The attributes of the Equipment object are defined in Table 4.5.

Table 4.5 Equipment Object Definition Table

Attribute Name	Definition	Access	Requirement	Form
ObjType	Object type.	RO	Y	Text = “Equipment”
ObjID	User-definable name.	RW	Y	Text

4.5.2 *TransferJob Object* — The attributes of the TransferJob object are defined in Table 4.6. A TransferJob is created by the equipment when it accepts a transfer job create request from the host, and its attributes are based upon that request.

Table 4.6 TransferJob Object Definition Table

Attribute Name	*Definition*	*Access*	*Requirement*	*Form*
ObjType	Object type.	RO	Y	Text = “TransferJob”
ObjID	Textual version of TRJobID.	RO	Y	Text
TRJobID	Identifier assigned by the equipment.	RO	Y	Unsigned integer
TRJobName	Host-created job identifier.	RO	Y	Text
TRJobState	Current state.	RO	Y	Enumerated: TRJobQueued, TRJobActive, TRJobComplete
TRAtomicList	List of (TRAtomicID).	RO	Y	List of unsigned integers

4.5.3 *AtomicTransfer Object* — The attributes of the AtomicTransfer object are defined in Table 4.7. AtomicTransfer objects are created by the equipment when it accepts a transfer job create request from the host, and its attributes are based upon that request.

Table 4.7 Atomic Transfer Object Definition Table

Attribute Name	*Definition*	*Access*	*Requirement*	*Form*
ObjType	Object type.	RO	Y	Text = “AtomicTransfer”
ObjID	Textual version of TRAtomicID.	RO	Y	Text
TRAtomicID	Identifier assigned by the equipment.	RO	Y	Unsigned integer
TRAtomicState	Current state.	RO	Y	Enumerated: TRAtomicActive, TRAtomicComplete
TRLink	Assigned by the host to identify an atomic transfer.	RO	Y	Unsigned integer
TRPort	Material I/O Port identifier.	RO	Y	Unsigned integer
TRObjName	Textual identifier of transfer object.	RO	N	Text
TRObjType	Material type of transfer object.	RO	N	Enumerated
TRRole	Equipment’s role.	RO	Y	Enumerated: Primary or Secondary
TRRecipe	Recipe identifier for transfer recipe.	RO	N	Text
TRPartner	Name of equipment’s partner.	RO	Y	Text
TRPartnerPort	Identifier for partner’s port.	RO	N	Unsigned integer
TRDirection	Indicates whether the equipment will send or receive.	RO	Y	Enumerated: Send Receive
TRType	Indicated the equipment is to be active or passive.	RO	Y	Enumerated: Passive Active
TRLocation	Identifies location on the equipment of the source (or destination) of the transfer object.	RO	N	Unsigned integer
TRAutoStart	Indicates whether the equipment may automatically start the handoff without waiting for a StartHandoff command.	RO	Y	Boolean: Automatically start (TRUE). Wait for StartHandoff (FALSE).

5 Micro Level

5.1 *Micro Level Concepts* — The micro level of material movement is defined below to allow for interaction between transfer partners on a step–by–step basis during a handoff. It was designed to be used in conjunction with the macro level defined above but may instead be used with some other transfer coordination mechanism.

5.1.1 *Micro Level Definitions*

Handoff — The process by which the transfer object moves from the sending transfer partner to the receiving transfer partner. The terms "handoff" and "micro level" refer to the same transfer activity.

Micro Move — One of a sequence of moves required during a micro level process to effect the transfer of an object. This is normally associated with physical activity within the transfer envelope.

5.1.2 *Micro Level Description* — While the macro level of material movement deals with coordination and preparation for a transfer, the micro level addresses the physical synchronization between two transfer partners during an atomic transfer. To control these physical interactions properly, communication between the transfer partners is required. Using these communications, the partners shall ensure that both are ready to perform the same specific transfer, that both agree on the mechanism, and that movement within the transfer envelope is properly coordinated.

There are three parts to the micro level interaction between transfer partners. Each of the three parts must occur in some form for the micro level to integrate properly with the macro level.

1. The first part is the exchange of "I'm ready" type messages, so that the partners can be sure that starting the transfer is appropriate.
2. The second part of the micro level interaction is the coordination of the physical transfer. During a transfer, control of the transfer envelope may belong to only one partner at any time. Joint or parallel movement is never allowed in this space[13]. The transfer itself is a series of steps called micro moves. Each step typically contains all the activities one partner can perform before the other must act. If the secondary partner is passive during the move, no micro level communications are needed for part two, since the primary partner shall control the transfer envelope during the entire physical transfer.
3. The third part of the micro level interaction is the verification that the transfer is complete.

The micro transfer mechanism assumes that the secondary partner will recognize and respond to some list of micro commands known to the primary partner. These commands might range from the very specific (e.g., "reach out and grasp the transfer object") to the very general (e.g., "perform the next step in your sequence").

5.2 *Micro Level Behavior*

5.2.1 *Micro Level Communications* — This section provides baseline definitions of the communications needed by the micro level of material movement. More detailed definitions of the messaging services can be found in Section 5.4 below. Refer to Figure 5.1 while reading the message definitions. In the figure, the messages in *italics* are macro level messages provided for reference. The **bolded** messages are the micro level messages.

The prefix "HO" (for **H**and**O**ff) is appended to the names of micro level messages to differentiate them from the "TR" transfer messages.

<u>HOReady</u> — Upon completion of any setup operations, each transfer partner shall declare to the other that it is prepared for the specified atomic transfer. Either partner may make this declaration first. When the HOReady exchange has been made, the primary partner shall start the transfer. The HOReady messages include a transfer identifier (see TRLink), which the partners match up. Looking back at the macro level atomic transfer state diagram (Figure 4.6), the combination of having sent and received a HOReady message for the specific transfer will spur the transition from the SETUP to AWAITING TRANSFER PARTNER states.

It is possible that an equipment will receive a HOReady message from its partner before it has received the corresponding Transfer Job Create request from the host (e.g., setup is instantaneous). An equipment shall be able to retain such a received message for a time to allow the host to deliver the corresponding Transfer Job Create request.

<u>HOCommand</u> — The HOCommand is the mechanism by which the primary transfer partner orchestrates the sequence of micro moves necessary for the transfer. The primary transfer partner shall successively perform its own micro move, then direct its partner to perform its next move. Thus, the partners may alternate control of the transfer envelope during the transfer. If the secondary partner is passive during the entire transfer and the primary performs all physical activity, no HOCommands are required — they are implicitly

13 The space outside the transfer envelope is not restricted, and some preparations for the next micro move may be possible in this space.

executed by the primary partner. Only one HOCommand may be active at any time.

The list of commands to execute may be included in a transfer recipe. See Section 4.2.4 above for more detail.

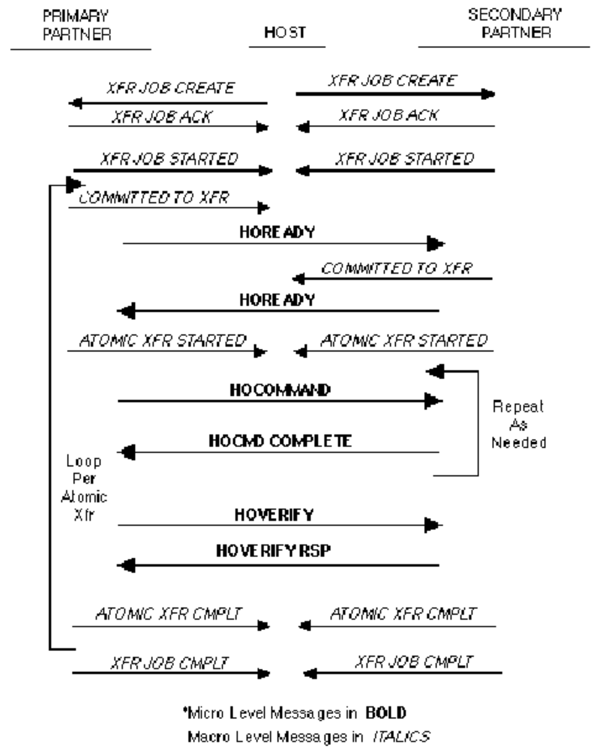

Figure 5.1
Micro Level Message Flow

When the secondary partner is passive, no HOCommands need be issued. In this case, all micro moves are conceived of and executed by the primary partner. The HOReady and HOVerify portions are still required in this case.

HOCommand Complete — Once the secondary partner has completed the task defined in the HOCommand, it uses the HOCommand Complete response message to alert the primary partner. If an HOCommand cannot be accepted or fails, this message shall be used to convey that information.

HOVerify — The primary transfer partner, believing all micro moves complete, shall check its port to ensure that the transfer object has been sent (or received) and that all mechanisms are in a safe position. Next, it shall send the HOVerify message to the secondary partner, asking that it perform the same check.

HOVerify Response — The affirmative response from the secondary partner confirms that the transfer was a success. This message may alternatively signify (through a data field) that the secondary partner does not consider the transfer to be complete. This message coincides with the transition for both partners from the macro level TRANSFERRING state to the ATOMIC TRANSFER COMPLETE state.

5.2.2 *Micro Level Messaging via Host* — As mentioned above, all three parts of the micro level of communications must be carried out in some form. If a direct link from equipment to equipment exists, they should occur via that means, whether using SECS–I, Parallel I/O, or another protocol. If no direct link exists, there is a means by which the host may relay micro level messages from one transfer partner to the other. This is illustrated in Figure 5.2.

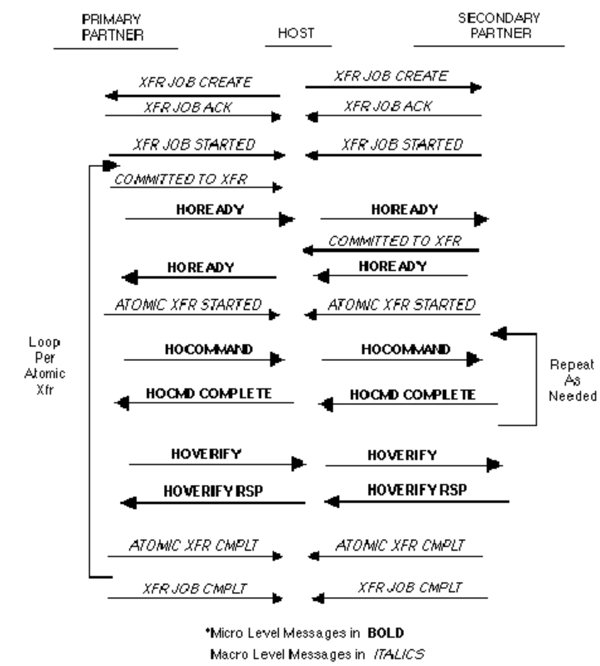

Figure 5.2
Micro Level Message Flow via Host

In this situation, the host may act as a surrogate of each equipment's transfer partner. A piece of equipment would send the same defined micro level message as before, but would deliver it to the host via the standard host link. The equipment would treat the host as its transfer partner, expecting the same response as if it were talking directly to its true transfer partner. The host is responsible for managing the relay of messages from one partner to the other as necessary.

The equipment shall provide the ability to pass micro level messages through the host. If a direct peer–to–peer link is available, the user shall be able to configure the path for the messages (directly to partner or through the standard host link).

5.3 *Extended Behavior* — This section discusses two micro level messages which may be exchanged between the transfer partners as a result of exceptional conditions.

HOCancelReady — Once a transfer partner has sent a HOReady message, it is committed to participate in a transfer. Under some circumstances, an equipment needs to withdraw from that commitment. The HOCancelReady message is used for this purpose.

The HOCancelReady shall be accepted by the transfer partner unless it has already sent the corresponding HOReady. In cases where both transfer partners have sent the HOReady message, the atomic transfer is considered to have begun and cannot be canceled. If the transfer partner receives an HOCancelReady, but has no record of receiving the original HOReady, it should accept the HOCancelReady.

HOHalt — The HOHalt message is used by a transfer partner in situations when the equipment or transfer object are endangered. The receiver of the HOHalt message shall cease all movement related to the atomic transfer immediately. Manual intervention is required before the halted partner may resume movement.

5.4 *Micro Level Services* — This section expands upon the messages defined in Section 5.2.1. It defines the parameters of the messages and discusses the possible values.

5.4.1 *Service List* — The following messages are exchanged between transfer partners during an atomic transfer.

Service Name	*Description*	*Confirmed*
HOReady	Each partner sends this message when it is ready to begin the transfer process.	No
HOCommand	Message sent by the primary transfer partner to cause the secondary partner to perform an activity in the transfer envelope. The response to this message indicates completion of the activity or error.	Yes
HOVerify	Message sent by the primary transfer partner informing the secondary partner that all activities should be complete and asking for confirmation.	Yes
HOCancelReady	Message sent to rescind a previous HOReady message.	Yes
HOHalt	Message sent to stop all action by the transfer partner. This message ends the transfer and requires manual intervention at the halted equipment.	Yes

5.4.2 *Service Detail* — The tables below define the parameters for each service. In some cases, parameters have additional detail which is defined in a following section. These parameters are marked with the "*" character.

The following codes are used in the definition of the parameters (e.g., how each parameter is used in each direction):

"M" — Mandatory Parameter—must be given a valid value.

"C" — Conditional Parameter—may be defined in some circumstances and undefined in others. Whether a value is given may be completely optional or may depend on the value of other parameter.

"U" — User-Defined Parameter.

"–" — Not Used.

"=" — (for Response only) Indicates that the value of this parameter in the response must match that in the primary (if defined).

The column labelled "Form" is used to indicate the type of data contained in a parameter. The forms used in this document are defined below.

Unsigned Integer — May take the value of any non–negative integer. Messaging protocol may impose a limit on the range of possible values.

Enumerated — The parameter may take on one of a limited set of possible values. These values are generally given textual names, but may have simpler representations in protocol (e.g., numeric values).

Boolean — The parameter may take on one of two possible values, equating to "True" and "False."

Text — A text string.

Structure — A complex structure which consists of a collection of values of one of the possible forms. The breakdown of all "Structure" parameters is provided within this document.

HOReady

An equipment indicates to its transfer partner a readiness to begin a specific atomic transfer.

Parameter	*REQ/IND*	*Description*	*Form*
EQName	M	Identifier (Name attribute) of the equipment sending this message.	Text
TRLink	M	Used by the transfer partners to assure that their communications relate to the same atomic transfer.	Unsigned integer
TRPort	M	Identifier of the port (on the equipment sending this message) to be used for the transfer.	Unsigned integer
TRObjName	C	Identifier of the object to be transferred.	Text
TRObjType	C	Type of transfer object, required if transfer might involve multiple types.	Enumerated
TRRole	M	Is the sender of this message the primary or secondary transfer partner.	Enumerated
TRPartner	C	Identifier (Name attribute) of the equipment expected to receive this message.	Text
TRPartnerPort	C	Material I/O port on the equipment which receives this message to be used for this atomic transfer.	Unsigned integer
TRDirection	M	Does the sender of this message expect to send or receive the transfer object.	Enumerated
TRType	M	Is the transfer to be passive or interactive?	Enumerated
TRLocation	C	Identifier of the material location (on the sender of this message) which will be the source (or destination) of the transfer object.	Unsigned integer

HOCommand

The primary transfer partner directs the secondary transfer partner to perform an action related to a specific atomic transfer.

Parameter	*REQ/IND*	*RSP/CONF*	*Description*	*Form*
TRLink	M	M(=)	Identifier of the atomic transfer which related to this HOCommand.	Unsigned integer
HOCmdName*	M	-	Specific command to be executed.	Text
(List of) CmdParameter	C	-	Parameter(s) related to the specified HOCommand. (May be a list of 0.) See Section 4.4.2 for definition of CmdParameter.	Structure
HOStatus	-	M	Reports success or failure of the command and, if failure, supplies error detail.	Structure

HOStatus Detail

Parameter	*RSP/CONF*	*Description*	*Form*
HOAck	M	Did the activity succeed or fail, was the inquiry accepted or rejected. True is a positive result, False is negative.	Boolean
(List of) Status	C	Reports any errors found. (May be list of 0) See Section 4.4.2 for details of this structure.	Structure

HOVerify

The primary transfer partner indicates completion of the atomic transfer to the secondary partner and requests confirmation of a successful transfer.

Parameter	*REQ/IND*	*RSP/CONF*	*Description*	*Form*
TRLink	M	M(=)	Identifier of the atomic transfer.	Unsigned integer
HOStatus	M	M	Reports the completion status of the atomic transfer from the standpoint of the sending transfer partner.	Structure

HOCancelReady

Having previously sent an HOReady message, a transfer partner determines that it is no longer ready to begin the transfer and sends the HOCancelReady message.

Parameter	*REQ/IND*	*RSP/CONF*	*Description*	*Form*
TRLink	M	M(=)	Identifier of the atomic transfer.	Unsigned integer
HOCancelAck*	-	M	Acknowledge code for HOCancelReady.	Enumerated

HOHalt

This message may be sent by either transfer partner when the equipment or transfer object is endangered. It requires the receiver to cease all transfer-related movement immediately. Manual intervention is required before the halted partner may again be available for transfer.

Parameter	*REQ/IND*	*RSP/CONF*	*Description*	*Form*
TRLink	M	M(=)	Indentifier of the atomic transfer.	Unsigned integer
HOHaltAccept	-	M	True if the request is accepted. Request may be denied only if referenced atomic transfer does not exist.	Boolean

5.4.3 *Parameter Definitions* — This section gives further detail on certain parameters defined above. Parameters which are defined in Section 4.4 are not repeated here.

ErrorCode

HOCommand:

- Rejected/Unrecognized Command.
- Rejected/Parameter Error.
- Failed (Completed Unsuccessfully) — indicates that the activity was attemped, but failed; the mechanisms continue to be operational.
- Failed (Unsafe) — indicates that the equipment was left in an unrecoverable state at the end of the activity due to the failure of a mechanism or due to the risk of damage to the equipment or material. Manual intervention is required.

HOVerify:

- Sensor-Detected Obstable.
- Material Not Sent.
- Material Not Received.
- Material Lost.
- Hardware Failure.
- Handoff Canceled Externally (by host or operator).

HOCmdName — The set of micro commands is provided below as an example. The commands used are not limited to this set. However, adherence to a common set of commands will speed implementation of material transfer.

This particular example applies to a situation where the secondary partner is active and the primary partner is passive.

Cmd	*Description*
Pick	Commands the secondary partner to take all actions necesary to remove the transfer object from the primary partner's port.
Place	Commands the secondary partner to take all actions necesary to place the transfer object in the primary partner's port.
Extend	Commands the secondary partner to reach into the primary partner's port area in close proximity to the material location.
Acquire	Commands the secondary partner to acquire control of the material.
Release	Commands the secondary partner to release control of the material.
Retract	Commands the secondary partner to withdraw from the primary partner's port area.
Store	Commands the secondary partner to place the transfer object in its final location.
Retrieve	Commands the secondary partner to pick up the transfer object from its initial location.
MoveTo	Commands the secondary partner to move its transfer mechanism (e.g., robot gripper) to a specified position (given as a parameter).

The Pick command might actually be an equivalent of the sequence:

Extend, Acquire, Retract, Store.

The Place command might be an equivalent to the sequence:

Retrieve, Extend, Release, Retract.

HOCancelAck — Conveys the response to the HOCancelReady request. Possible values are:

0 — Accept Cancel.

1 — Atomic Transfer Unknown.

2 — Reject Cancel—Transfer Begun.

HOHaltAck — Conveys the response to the HOHalt request. Possible values are:

0 — Accept Halt.

1 — Atomic Transfer Unknown.

APPLICATION NOTES

A1-1 *Transfer-Related Mechanisms* — A transfer job interacts very closely with port-related hardware. This may include a variety of mechanisms such as port doors, cassette indexers, robot arms, carrier restraints, etc. The state models for the transfer job and atomic transfers shown above relate to this port hardware, but do not define hardware behavior directly. Therefore, an additional state model is needed for the port hardware.

A1-1.1 *Example Port State Model* — Figure A1-1 provides an example of a state model for a port. It is divided into three AND substates: MECHANISM, AVAILABILITY, and PAUSE. These substates represent the three most common areas of interest to the host.

Below, the states shown in Figure A1-1 are described, followed by a transition table. This is presented as an example. The actual model is implementation-dependent.

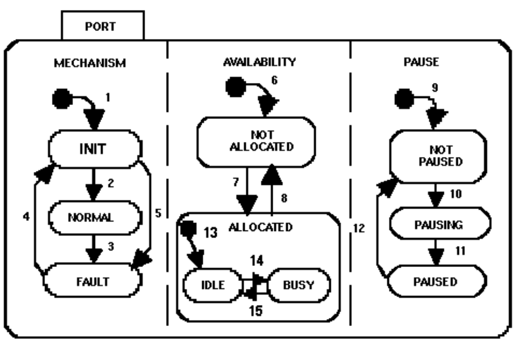

Figure A1-1
Example Port State Diagram

MECHANISM

This state pertains to the port's overall ability to function. It has three substates: INIT, NORMAL, and FAULT.

INIT

This is the default state upon powerup or reset of the port. It includes some unspecified operations to ensure that the port is ready to operate.

NORMAL

When in this substate, the port is in normal working order.

FAULT

In this substate, a problem has been detected with a port mechanism.

AVAILABILITY

This state pertains to the availability of the port to perform a function. This state has two substates: NOT ALLOCATED and ALLOCATED.

NOT ALLOCATED

In the NOT ALLOCATED state, neither the equipment nor the host has allocated the port. The port is waiting to be allocated to either a transfer job or to some process-related task. This is the default state upon equipment system startup.

ALLOCATED

In this state, the port is allocated to either the equipment (typically process-related) or to the host for a transfer job.

NOTE: If the port is allocated to an equipemnt function and the host does not wish to wait for the equipment to release the port, it must take some other action to force the port to become available. One possibility is a “Pause for Transfer” remote command that might be used during a processing cycle.

IDLE

Port is allocated to a task (e.g., processing or transfer job), but activity on the task has not yet begun.

PAUSE

This state pertains to the pausing of current port activity. Pausing of the port is different than pausing a transfer job which might be using the port.

NOT PAUSED

The port defaults to NOT PAUSED upon system startup. In this state, port activities are not hindered by the pause function.

PAUSING

In this state, the port has received a directive to pause and is in the process of bringing the current activity to a halt.

PAUSED

All port activity has ceased. The port is awaiting a directive to resume activity.

Table A1-1 describes the transitions for the example port state model. In the application-specific implementation, events should be defined for all significant occurences related to the model.

Table A1-1 Extended Atomic Transfer State Transition Table

#	*Current State*	*Trigger*	*New State*	*Action*	*Comment*
1	(Undefined)	Entry into PORT Model.	INIT	None.	None.
2	INIT	Port mechanisms found to be functional.	NORMAL	None.	None.
3	NORMAL	A malfunction of the hardware occurs.	FAULT	None.	None.
4	FAULT	The malfunction is "fixed." This may be a manual re-initialization of the port.	INIT	None.	None.
5	INIT	A problem occurs during initialization.	FAULT	None.	None.
6	(Undefined)	Entry into PORT Model.	NOT ALLOCATED	None.	None.
7	NOT ALLOCATED	The port is allocated to transfer or process job.	ALLOCATED	None.	None.
8	ALLOCATED	Port is released from current job.	NOT ALLOCATED	None.	None.
9	(Undefined)	Entry into PORT Model.	NOT PAUSED	None.	None.
10	NOT PAUSED	Directive issued to pause port.	PAUSING	None.	None.
11	PAUSING	Port activity has ceased.	PAUSED	None.	None.
12	PAUSED	Directive issued to resume activity.	NOT PAUSED	None.	None.
13	(Undefined)	Entry into ALLOCATED state.	IDLE	None.	None.
14	IDLE	Task activity begins.	BUSY	None.	None.
15	BUSY	Task activity ends.	IDLE	None.	None.

A1-1.2 *Additional Related Mechanisms* — Many equipment material ports have additional related mechanisms that can aid or interfere with a transfer. These mechanisms can be represented as parallel (or orthogonal) states within the port state model. Most such state models will be simple two–state instances (e.g., open/closed, on/off).

In some circumstances, the host will have a need to be able to understand the status of these mechanisms and may wish to be informed of state changes. This is especially true if the host is to do error handling and preliminary diagnosis of problems. The host also may need some level of remote commands to manipulate these mechanisms. Some possible transfer-related mechanisms are discussed below.

Port Material Sensors: These sensors detect the presence or absence of material in a port. The change in this status is important, and an event should be provided to report change in this condition.

Port Doors: Some equipment closes the environment internal to the equipment, except during material transfer. The applicable states are normally open/closed.

Port Elevator: Port elevators have multiple states such as HOME, MOVING, SLOT 5 ACCESS, or interlocks such as BLOCKED BY INTERNAL TRANSFER ARM.

Robot Arm: The transfer arm may be in motion, may be holding material, etc.

Carrier Restraints: On some ports, carriers are clamped in place and cannot be removed unless the restraints are released. There would be a clamped/unclamped state model.

A1-2 *WIP Tracking* — WIP tracking refers to the ability of an entity (host or equipment) to keep accurate track of the material within its domain. Typically, the equipment is expected to track the material's location within its modules and subsystems, while the host may take a more global view, knowing at which equipment (or storage area) a specific transfer object resides.

The process of material movement crosses the boundary of the equipment domain. Therefore, there must be information exchanged between host and equipment about what material is being transferred and what the plans are for the material (e.g., when and how processed).

It is outside the scope of this standard to define WIP tracking capabilities, either at the host or equipment level. However, it is critical for the material movement function to supply enough information that WIP tracking is feasible. Specifically, the material movement messaging must identify the object being transferred. WIP references are kept at a minimum. For example, for a carrier containing wafers, the material movement process would need to know the carrier identifier, but need not be concerned with the specific lot contained within. It is also outside the scope of this standard to be concerned with the detailed contents or status (e.g., processed or not) of a transfer object. These are important issues that should be dealt with elsewhere.

A1-3 *Error Handling* — This section addresses some common problems that may occur during the transfer process. This is not intended to present hard and fast rules for error handling, but rather to raise potential issues and suggest possible solutions. Several primary error or interrupt types are covered. For state model representations of hardware faults, see Section A1-1.

NOTE: Transfer errors are not completely communicated between transfer partners. The host may be responsible for pausing or aborting (as needed) the activities of the partner or equipment with an error condition.

A1-3.1 *Macro Level Error Handling*

1. Transfer Specification Error

This would consist of an error in the data contained in the Transfer Job Create request. Some examples might include specification of a static port as an active transfer partner or the use of a non–existent port on the equipment. Such an error should result in rejection of the Transfer Job Create request.

2. Setup Error

It is during the setup for an atomic transfer that many prerequisites are typically checked. A setup error may be either a hardware or a transfer specification problem. Hardware problems might be such events as cassette elevator fault, door fails to open, etc. If a hardware problem occurs, the transfer job is paused and the hardware state model makes a transition to the FAULT state (see Section A1-1). When the problem is corrected, the host may resume the activity or stop the transfer job.

A transfer specification problem would typically relate to the presence or absence of material at specified locations. If there is a transfer specification problem found, a transition to the RESTORE state is made (transition #5) and the transfer job is terminated.

3. Restore Error

A restore error occurs when the port cannot be brought back to its desired physical condition for the NOT ALLOCATED state. This would be a failure of a hardware mechanism. The hardware should then transition to the hardware FAULT state (see Section A1-1). The "completeness" of the transfer job may not be hindered by problems in the restore activities. However, the next transfer job may be affected.

4. Transfer Error

Transfer errors may be of many types and of varying levels of severity. Any error that occurs must be reported to the host. If the error is caused by failure of a port mechanism, a transition to the hardware FAULT state should occur (see Section A1-1). The problem might also stem from a problem with the transfer specification or in the transfer recipe (i.e., specification is legal, but does not match that of the partner). If failure in the transfer hardware has occurred, the transfer job should transition to the PAUSED state and report it to the host. The host may choose to abort the transfer job, or correct the problem and resume the transfer job.

5. Timeouts

To ensure that all transfer problems are detected, the host should implement a process for timeout monitoring of transfer jobs. The timeout periods need only be very rough estimates of the actual times required for the transfer job (e.g., 2× or 3× actual). The host might choose to monitor individual atomic transfers if tighter tolerances are required. The goal is to ensure that problems are not left undetected over extended periods.

To determine the proper time to allow for a transfer job, the host would sum the estimated maximum times needed for the individual atomic transfers which make up the job[14] . The atomic transfer time estimates would typically be determined by experience, since they rely on the interactions of two different transfer partners. A doubling or tripling of the average observed transfer time would be a good estimate. In some cases, suppliers of dedicated transfer equipment may be able to supply reasonable estimates.

Transfer job timeout monitoring should begin with the receipt of the "Transfer Job Started" message and end with the "Transfer Job Complete" message (both via the TRJobAlert service). If the host is monitoring the atomic transfers, monitoring should begin at the "Atomic Transfer Started" event and end at the "Atomic Transfer Complete" event (both via the TRJobEvent service). The host must ensure that reporting of these events is enabled in this case.

If a transfer job exceeds the timeout period, the host may either pause the transfer job or attempt to end it with the stop command. The former may be preferred, since human intervention is required in either case and may enable a resume command to continue the transfer.

A1-3.2 *Micro Level Error Handling*

1. No Communication With Transfer Partner

If a transfer partner attempts to communicate with its transfer partner and finds that the communications link is not responding, it should inform the host via an event

14 This assumes that atomic transfers proceed sequentially. If they may proceed in parallel, the sequence for each port may be summed and the longest total chosen as the estimated time.

and continue to retry at intervals. Note that a disconnect may be normal in some cases where the link is only made when the transfer partners are in close proximity.

2. TRLink's Not Matched

If two transfer partners are attempting different handoffs using the same physical resources (locations, mechanisms, etc.), the host should be informed via an event. The partners shall continue to await an appropriate HOReady message—allowing the host to "fix" one or the other of the partners. If an equipment has no current transfer and it receives a HOReady message, it should hold that message until a transfer job is defined for it.

3. HOCommand Not Accepted

If the secondary transfer partner cannot accept a micro command from the primary partner, it should use the HOCommand Response message to inform the primary partner that the command was rejected. The primary partner may affect some recovery or inform the host that the transfer has failed.

4. HOCommand Failed in Execution

If the secondary transfer partner fails in the execution of an HOCommand, it should notify the primary partner of the problem. The primary partner may affect some recovery or inform the host that the transfer has failed.

5. Transfer Not Confirmed

If the secondary transfer partner sends a negative HOVerify Response message saying that the transfer object has not been properly transferred, the primary partner should not attempt recovery, but inform the host that the transfer has failed.

6. Timeout Awaiting Partner Readiness (HOReady)

Timeout Awaiting Micro Command Completion (HOCommand Response)

Timeout Awaiting Transfer Verify Response (HOVerify Response)

In some cases, action may be required if a transfer partner has been awaiting communication from its partner. The equipment should monitor timeout periods for each of the three cases. The factory user may need to customize the timeout periods for his/her factory. When a timeout occurs, the equipment should notify the host (via a TRJobEvent message) that the situation has occurred, reset the timer, and continue to await the partner. The host is responsible for taking any further action which is required (e.g., Stop or Pause command).

A1-4 *Practical Applications* — This section provides examples of practical applications of the material movement services as presented in this document. Four examples are provided below. Within these are presented some major variations expected in a factory situation. These are not exhaustive in scope, but the expectation is that reviewing these examples will help the reader understand how to apply these concepts to a real application. These examples are also not intended to represent a cohesive implementation strategy, but rather some alternatives that might be considered. AGVs are used in the examples below because they are familiar to most readers, not because they are favored over other transfer agents. To simplify the explanation, the examples below deal only with movement of cassettes.

A1-4.1 *Passive Transfer* — In this example, a transfer is to be performed between an AGV with robot arm and a piece of processing equipment. A cassette is to be picked up by the AGV for later delivery to another piece of equipment. The equipment has a cassette indexer and a port door, but no other port-related mechanism that would either aid or hinder transfer. The chosen transfer mechanism provides for the equipment to act as a passive transfer partner with no micro level communications required. This is called a passive transfer. The primary transfer partner is the AGV.

The setup for the equipment is to check to make sure that the correct material is in the specified port, drive the cassette elevator to the home position (if not already there), and open the port door. Note that the first two setup activities could have been designed to occur as a part of the micro level transfer if desired. The setup operations for the AGV are to ensure that its receiving port is empty and to drive to the transfer position in front of the designated equipment.

The transfer proceeds in the following steps:

1. The host sends a Transfer Job Create message to each transfer partner.
2. The partners each accept the transfer job.
3. The equipment both have the needed available resources so the transfer job immediately. Thus, Transfer Job Started messages are sent to the host by each.
4. Each partner in turn completes its setup activities for the atomic transfer, sends a "Committed To Transfer" event to the host, and then sends an HOReady message to its transfer partner (via the host).
5. Each partner sends the "Atomic Transfer Started" event to the host as soon as it determines that it has both sent and received an HOReady message for this handoff.

6. The AGV (as primary transfer partner) begins the transfer.
7. The AGV performs the transfer. Notice that no HOCommand messages are needed since this is a passive transfer.
8. The AGV sends an HOVerify message to the equipment via the host.
9. The equipment ensures that the cassette is no longer sensed in its port and then sends an HOVerify Response message (via the host) to the AGV, followed by an “Atomic Transfer Complete” event to the host.
10. Upon receipt of the HOVerify Response message, the AGV sends an “Atomic Transfer Complete” event to the host.
11. Each partner completes its RESTORE operations and then sends a Transfer Job Complete message to the host.

The transfer is now complete. The host may now direct the AGV to deliver the cassette to a new destination.

A1-4.2 *Active Transfer/Exchange of Cassettes* — This example addresses the situation where the factory control system needs to remove a processed lot from an equipment and immediately replace it with an unprocessed lot. The assumptions are that the processing of the lot on the equipment is nearing completion and that the AGV has already acquired the unprocessed lot that will next be placed on the equipment. Direct micro level communications exist between the AGV and the equipment. This will be an interactive transfer.

The setup operations for the equipment are opening the port door, driving the cassette indexer to the home position, and checking for the presence of the proper cassette. AGV setup is movement to the transfer location for that equipment. During the transfer, the equipment acts as the primary transfer partner, and will unclamp/clamp the cassette (clamps hold the cassette in the proper position). The AGV will interact with the equipment as the secondary transfer partner.

The transfer proceeds as follows:

1. The host sends Transfer Job Create requests to the equipment and to the AGV. The timing is chosen so that the time required for the AGV to travel to the equipment is approximately equal to the time left to complete processing of the lot to be transferred. The Transfer Job Create message to each contains two atomic transfers, the first dealing with the removal of the processed cassette from the equipment and the second dealing with the loading of the unprocessed cassette onto the equipment.
2. The AGV accepts the transfer job and begins the first atomic transfer immediately, sending a Transfer Job Started message to the host. Setup begins — the AGV begins traveling toward its transfer partner. The equipment accepts the transfer job and retains it for later execution.
3. When processing completes on the lot, the equipment begins the transfer job. It sends a Transfer Job Started message to the host and begins its setup operations for the first atomic transfer.
4. The AGV completes its setup activity and sends a “Committed To Transfer” event to the host. It also sends a HOReady message to the equipment.
5. The equipment completes its setup, then sends a “Committed to Transfer” event to the host and an HOReady message to the AGV.
6. Since the AGV had previously declared itself ready, the equipment sends an Atomic Transfer Started event to the host and starts the transfer.
7. The equipment begins by sending an HOCommand that results in the AGV reaching out and grasping the cassette.
8. Upon receiving the Command Complete message from the AGV, the equipment unclamps the cassette, allowing its removal.
9. The equipment sends an HOCommand that results in the AGV removing the cassette from the equipment. The material sent and material received events are sent to the host by the respective partners.
10. When the equipment receives the Command Complete message from the AGV, it considers the transfer to be complete. It sends the HOVerify message to the AGV.
11. The AGV sends the HOVerify Response message to the equipment.
12. Each of the transfer partners now determines that another atomic transfer is required in order to complete the transfer job.
13. Each transfer partner now transitions to the second atomic transfer. They each send “Atomic Transfer Complete” events.
14. Each transfer partner now begins the new setup phase. The AGV determines that it has already reached the transfer point and has the unprocessed cassette to be transferred. The equipment determines that the port is now empty, and that the door is open. This setup time was saved.

15. The second atomic transfer now proceeds in a similar fashion through steps 4–12 above. That done, since there are no further moves required, the restore stage begins, including "Atomic Transfer Complete" events to the host.

16. Once restore operations are complete, each sends a Transfer Job Complete message to the host. The transfer is now complete.

A1-4.3 *Sequential Moves* — In this example, the factory control system desires to move a cassette from equipment "A" to equipment "B" using a fixed robot stationed within reach of the two. Each piece of equipment has a static port and will act as passive transfer partners in all cases. The robot will act as the primary partner for all transfers. The host system must do the high-level planning of this move, determining that the lot to be transferred will be ready to go, that the robot is available, and that the receiving equipment is available to receive the cassette.

To accomplish the transfer, two atomic moves are required. First, the robot picks up the cassette at "A." Second, the robot deposits the cassette on "B." The gripper on the robot is considered its port in this case, since that is the point at which ownership of the cassette changes.

Since the step–by–step transfer details are similar to the first two examples, they will be omitted. However, a higher-level sequence will be given.

1. The host will send Transfer Job Create messages to "A," "B," and the robot. The robot will be given two sequential atomic moves to perform, the equipment one each. Note that the Transfer Job Create request to "B" could be delayed, if desired, until the completion of the first atomic move.

2. The first atomic transfer is performed: the robot removes the cassette from "A." Note that no HOCommands are issued, since the secondary partner is passive.

3. The second atomic transfer is performed: the robot places the cassette on "B."

4. Both "A," "B," and the robot deliver Transfer Job Complete messages to the host.

The transfer is complete.

An intelligent robot control system could combine the two atomic moves discussed above into a single robot program, achieving a more efficient transfer (e.g., no redundant movement during the transfer). In such a case, both "A" and "B" would need to be ready for transfer before the robot program was executed. The only limitation is that this combination would have to match the model and messaging discussed above.

A1-4.4 *Linked Lithography* — The final example is one where the micro level of material movement might be used without the macro level messaging described in this document. The linked lithography system described below is not patterned after any existing system and may not be the optimal implementation for such a system.

For this example, the challenge is to transfer wafers from a typical track system (following the coat/bake process) into a stepper for imaging, and then back to the track system for the develop process. In this example, the stepper acts as a slave to the track system, accepting wafers as they are given. The non–transfer-related communications will not be discussed.

In the wafer transfer process, the track takes on the macro level duties of the host. It is responsible for informing the stepper that a wafer is to be transferred, which wafer, and by what path and mechanism. It will initiate the transfer in a method analogous to the Transfer Job Create request. There is no transfer agent in this case. Either equipment might act as the primary transfer partner — it is irrelevant to this example.

The stepper and track will then exchange the normal micro communications. First, the HOReady messages will be sent. Next, the track will issue HOCommands to cause the wafer to be physically transferred. Finally, the HOVerify message transaction would occur. Meaningful events would be reported to the host as appropriate.

An alternative would be to design the system such that the track system tells the stepper system, in effect, "I'll be sending a series of wafers to you." Since the HOReady message contains sufficient information to describe the transfer (assuming a specific transfer recipe is not required), no specifics about the individual transfers need be exchanged until the HOReady is sent. In such a scheme, the stepper would take the HOReady message as description of the specific transfer and respond with an HOReady to match.

A1-4.5 *Non–Compliant Transfer Partner* — There will be situations where only one of the transfer partners is compliant with this standard. It is reasonable to use this transfer methodology in some such cases, especially those involving passive transfer. The host would be responsible for emulating a compliant transfer partner on behalf of the non–compliant partner (for the benefit of the compliant partner). Examples of non–compliant transfer partners might include a WIP rack, a human transfer agent, an older process equipment, or any material handling systems that support other protocols.

NOTICE: These standards do not purport to address safety issues, if any, associated with their use. It is the responsibility of the user of these standards to establish appropriate safety and health practices and determine the applicability of regulatory limitations prior to use. SEMI makes no warranties or representations as to the suitability of the standards set forth herein for any particular application. The determination of the suitability of the standard is solely the responsibility of the user. Users are cautioned to refer to manufacturer's instructions, product labels, product data sheets, and other relevant literature respecting any materials mentioned herein. These standards are subject to change without notice.

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

Index

NOTE: The numbers in this index represent the section, table, and/or figure that contains the corresponding subject.