has not mounted any other Circuit Modules yet or it is on the way of fabrication.

9.5.3.1 *Mother Substrate Class Definition* — Mother Substrate inherits Circuit Module. Class definition of the Mother Substrate is completely same as one for Circuit Module. This class is redefined to separate class Die from the other Circuit Module classes for convenience.

9.5.4 *Wafer* — This is a type of Mother Substrates which represents thin disc shape semiconductor material to fabricate semiconductor devices on it. Examples are 300 mm diameter silicon wafers and 3 inches diameter gallium arsenide wafers. Some case a few hundreds of microprocessor dice may be fabricated on it.

9.5.5 *Glass Plate* — This is a type of Mother Substrats which represents thin transparent rectangular glass plate to fabricate flat panel display devices on it. However it is usually rectangle, sometimes it may be such other shape as circle. Some case length of one of ridge is beyond one meter.

9.5.6 *Dicing Frame* — This is a type of Mother Substrates which represents thin rectangular flame which centers a piece of elastic sticky film to keep a wafer for dicing. Dicing frame is usually reusable.

9.5.7 *Cassette* — This is a type of Mother Substrates which represents one of reusable container to store and carry Circuit Modules. Contained Circuit Modules are positioned. Examples are wafer carriers, wafer shipping boxes, dicing frame magazines and IC trays.

9.5.8 *Tray* — This is a type of Mother Substrates which represents a kind of cassettes which holds dice on two dimensional surface of it. It is often used to move dice to next process, ship or test.

9.5.9 *Lead Frame* — This is a type of Mother Substrates which represents base of a die to package as a semiconductor device. It has many electrodes to connect with electrode pads of the die. A few Lead Frames are often physically linked together before packaging is completed for convenience of production. In this sense Lead Frames are a kind of Strip described later.

9.5.10 *Tape* — This is a type of Mother Substrates which represents very long narrow strip flexible substrate made of plastic film, cloth, paper or hybrid material to hold electronics elements or small parts for carrying, feeding or shipping. Usually it is rolled on a magazine for handling.

9.5.11 *PCB* — Print Circuit Board. Hard or flexible substrate with one or multiple layered circuit wiring conductor patterns an active/passive electronics elements on one of or both surfaces. This is also a type of Mother Substrates.

9.5.12 *Strip* — Usually rectangular or tape shaped substrate to mount electronics parts. This is also a type of Mother Substrates.

9.5.13 *Cabinet* — An enclosure for electronics to keep from exposure of electrode and/or subcomponents, and/or to give some means of human operation. Often it contains PCBs and power supply module with switches and such display devices as light emission diodes and flat display panel on it. This is also a type of Mother Substrates.

10 Detail Concepts

10.1 *System Wide Concept* — Previous section describes basic concepts of Die Tracing. Based on these, this section describes more detailed concepts to trace dice practically.

10.2 *Die Trace System* — A system for die tracing ranges over any entity related to Die Trace Data of the target Die. From the system point view, there must be some managing objects to coordinate ensembles in the system to achieve expected responsibility.

10.3 *Die Trace System Linkage* — Also it is difficult to manage a lot of objects and data by just one manager. It is expected to have more than one manager to coordinate and manage data and such objects as Machines and Tracers. They link together to organize a kind of network as illustrated in the following figure.

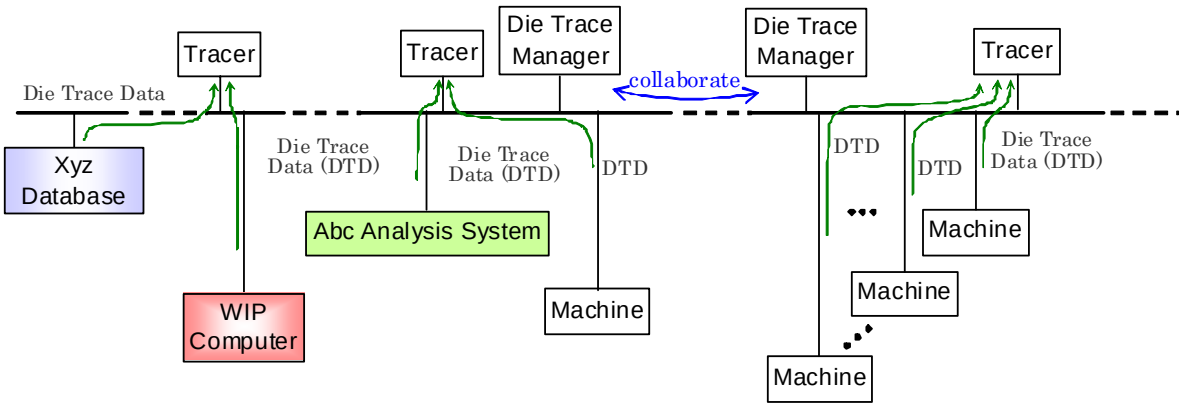

Figure 9
An Example of Die Tracing Manager Network

10.4 *Upgraded Ensemble Model* — The following diagram shows practical ensemble model with managing objects. A Die Trace Manager (DTM) coordinates, configures and does other managing responsibility to make whole the system work. For the other practical update is extensibility of Die Trace Systems. This model allows consolidation or connectivity of separated Die Trace Systems easily. Network of Die Trace Managers makes it possible to exchange die trace information widely and it gives scalability of Die Trace System.

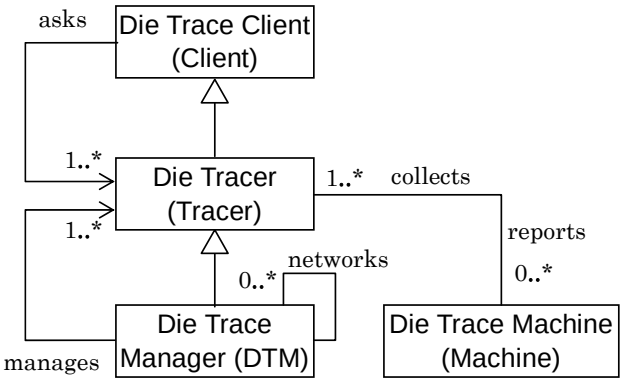

Figure 10
Die Trace System Model

10.5 *Class Definitions in the Model* — Classes appears in the Die Trace System Model are defined in the following subsections. Most classes have inheritance associations and inheriting classes has services of inherited classes.

10.5.1 *Die Trace Client (Client)* — This class represents client to ask something related to Die Tracing. Also this class represent system user so that the system may know problematic phenomena or notify alert about specific Circuit Modules.

10.5.1.1 *Responsibility of Die Trace Client* — Because this represents user end of Die Tracing System, it uses the other part of the system and it is also interface to the user. This class realizes the following things to complete the responsibility.

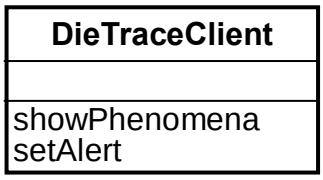

Figure 11
Die Trace Client Class

10.5.1.2 *Services of Die Trace Client* — This class provides the following services to complete its responsibility.

10.5.1.2.1 *showPhenomena* — This optional service returns problematic phenomena possibly happen on suspicious Circuit Modules registered on users system. The phenomena could be a part of Die Trace Data for a Die Tracer. The phenomena is given by any property to specify the suspicious Circuit Module or group of those and the other information about the phenomena.

10.5.1.2.2 *setAlert* — This optional service sets alert for a group of Circuit modules. The phenomena must be a kind of Die Trace Data and it may give information for the other part of Die Tracing System.

10.5.1.3 *Behavior of Die Trace Client* — Behavior is defined to complete responsibility of the Die Trace Client. The behavior is often specified through collaboration with outside of this class and state model.

Collaboration is reflected on some association of this class with others. The collaboration may happen when some service on this class is requested if the service needs some related service on any associating class to complete the original service. They are described in a part of service definitions above. State model is the other part of representation of the behavior of this class. Because this class is stateless, this class defines no state model.

10.5.2 *Die Tracer (Tracer)* — This class represents data collector for die tracing. The data come from Die Trace Machine or some other entities in the system.

10.5.2.1 *Responsibility of Die Tracer* — It keeps collected data somewhere in the system by any means. However it may have shared database with some other colleagues to keep the data, this document doesn't specify anything about that. How collected data are managed is out of the scope of this specification and this document define as much as possible for the case it has independent data management.

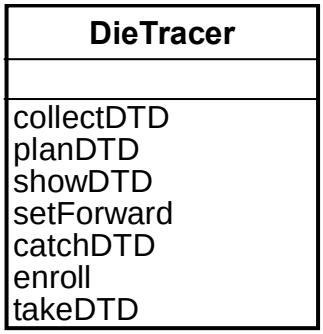

Figure 12
Die Tracer Class

10.5.2.2 *Services of Die Tracer* — Die Tracer provides the following services to accomplish its responsibility.

10.5.2.2.1 *collectDTD* — This optional service collects further Die Trace Data from data sources, if subordinates Die Trace Data source has the data, and replies. If something takes time to collect, it replies immediate data as much as it has and reports later when the data is available.

10.5.2.2.2 *planDTD* — This optional service provides a capability to set which data are required to be acquired. If anticipated data cannot be acquired because such data may not be supplied on specific equipment, it doesn't accept this request. It may take time if it needs to ask process machines whether it has the capability to provide specified data. Die Tracer replies information that the service request is acknowledged but detail possibility is notified later in such cases. This is an optional service and the means how to specify specific data item is out of scope of this document because it is dependent to equipment. If reported data are fixed by implementation, this service has no effect at all.

10.5.2.2.3 *showDTD* — This service replies requested Die Trace Data. Tracer may ask requested data to its data source before it replies when enough data is not there. If requested data is not on the tracer and it is difficult to acquire, it replies no data. If the data item is strange for the Tracer, it may ask peer Tracer acquainted or DieTraceManager.

10.5.2.2.4 *setForward* — This optional service forwards a part of or all Die Trace Data to specific URI. Forwarding data should be selectable by category, data type or originating machine. Also whether the forwarded data is left on the object or removed should be selectable. This service is optional and typically used for specific application or at the time data buffer is almost full. Interface to receive the data on the URI, means to send (e.g. SMTP and file transfer) from this object and application of the data on the URI are out of the scope of this document. If specified transferring means is not familiar with this service, an error will be returned.

10.5.2.2.5 *catchDTD* — This optional service receives forwarded Die Trace Data. Die Tracer shall handle the data without confusion with regular Die Trace Data coming from subordinate Machines. Usage, purpose and application of the data are out of the scope of this document.

10.5.2.2.6 *enroll* — This optional service registers specified Die Trace Manager as its supervisory Manager. Because the manager is only one for the tracer, this service is used to take over the role of the Die Trace Manager for the Tracer. This service doesn't affect if implementation fixes the Manager.

10.5.2.2.7 *takeDTD* — This service receives Die Trace Data from its subordinate Machines. This service is a notification service and it may be realized by event subscription of a kind of event channel facility for particular implementation.

10.5.2.3 *Behavior of Die Tracer* — Behavior is defined to complete responsibility of the Die Tracer. The behavior is often specified through collaboration with outside of this class and state model. Collaboration is reflected on some association of this class with others. The collaboration may happen when some service on this class is requested if the service needs some related service on any associating class to complete the original service. They are described in a part of service definitions above. State model is the other part of representation of the behavior of this class. Because this class is stateless, this class defines no state model.

10.5.3 *Die Trace Manager (Manager)* — This class represents management entity in Die Trace System and a kind of a broker among Die Tracers.

10.5.3.1 *Responsibility of Die Trace Manager* — This object manages all Die Tracers in Die Trace Management Cell which is the range of the manager. Die Trace Manager repeats asking all Die Tracers when inquiry about Die Trace Data. Also Die Trace Manager may participate a network of Die Trace Managers for more information exchange of Die Trace Data. When some information or inquiry comes, it may forward it to the other network members.

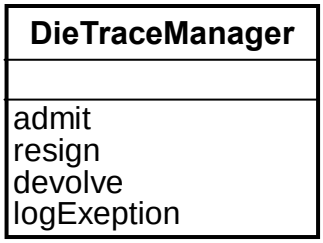

Figure 13
Die Trace Manager Class

10.5.3.2 *Services of Die Trace Manager* — Die Trace Manager provides the following services to complete its responsibility.

10.5.3.2.1 *admit* — This service accepts a Die Trace Manager to be a member of its network for Die Trace Managers. In the network Die Trace Data are shared even if they may be stored on different hardware entities or different network segments. When some inquiry comes to one of the member of the network, it broadcasts the inquiry to the other members if it has no corresponding data locally or there must be further data in the network. A member of the network who knows the data replies to the inquirer. If the data is private for its owner, no reply may be given or notice with negative reason may be responded. Some member may be located outside of intranet. Network security and authentication/authorization are outside of the scope of this document, and final users of this standard or network solution providers shall take care this issue to prevent serious problems.

10.5.3.2.2 *resign* — This service accepts a member Die Trace Manager to resign from Die Trace Managers network. If requested Die Trace Manager has not been a member, this service replies just error information.

10.5.3.2.3 *devolve* — This optional service accepts all or a part of Die Tracers transferred from a Die Trace Manager as subordinates of this Die Trace Manager. This manager ask a service of each devolved Tracer to enroll its subordinate. Before the enrollment has not been completed the Tracers may report Die Trace Data to previous Manager. The manager is recommended to forward the data to the new Manager.

10.5.3.2.4 *logExeption* — This optional service logs specified message for errors, peculiar events and other exceptions happened on each Die Tracer. The logged information shall be kept for some fair period and/or displayed on the console of the computer or its remote terminal window. This service is typically used for resource-less Die Tracers or concentrated exception monitoring.

10.5.3.3 *Behavior of Die Trace Manager* — Behavior is defined to complete responsibility of the Die Trace Manager. The behavior is often specified through collaboration with its outside and state model. Collaboration is reflected on some association of this class with others. The collaboration may happen when some service on this class is requested if the service needs some related service on any associating class to complete the original service. They are described in a part of service definitions above. State model is the other part of representation of the behavior of this class. Because this class is stateless, this class defines no state model.

10.5.4 *Die Trace Machine (Machine)* — This class represents production equipment to be source provider of Die Trace Data including process, metrology, test, material handling, assembly, packaging, mounting and any means used for manufacturing. Even handwork process could be the Machine as long as it records work and data during the work electrically, if such data are effective for Die Tracing purpose.

10.5.4.1 *Responsibility of Die Trace Machine* — This object reports expected Die Trace Data to Die Tracer.

10.5.4.2 *Services of Die Trace Machine* — Die Trace Machine provides the following services to complete its responsibility.

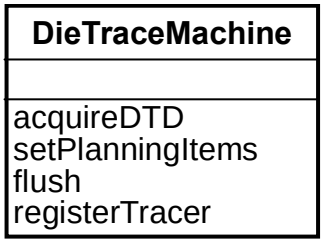

Figure 14
Die Trace Machine Class

10.5.4.2.1 *acquireDTD* — This service provides expected Die Trace Data on production equipment. It reports corresponding data immediately if the data has

been already available on the equipment. If not, it acquires the data before reporting. If the acquisition was impossible, no data is replied.

10.5.4.2.2 *setPlanningItems* — This service specifies expecting data items for die tracing. The items are chosen from potential and reportable data list for die tracing in specification exchanged between equipment vendor and user. Otherwise both party need to prepare a kind of metadata which describes reportable data items and their properties including data sources equipment itself electrically to allow user's application to specify expecting data item by looking up the metadata.

10.5.4.2.3 *flush* — This optional service let equipment send all planned data which have been already acquired but not sent yet. Same data will responded with acquireDTD service if all planned data items are specified. This service is typically used when the Tracer of this Machine is replaced.

10.5.4.2.4 *registerTracer* — This optional service allows switching corresponding Tracer dynamically.

10.5.4.3 *Behavior of Die Trace Machine* — Behavior is defined to complete responsibility of the Die Trace Machine. The behavior is often specified through collaboration with its outside and state model. Collaboration is reflected on some association of this class with others. The collaboration may happen when some service on this class is requested if the service needs some related service on any associating class to complete the original service. They are described in a part of service definitions above. State model is the other part of representation of the behavior of this class. Because this class is stateless, this class defines no state model.

11 Suggestive Complements

11.1 *Illustrative Models in Production Line* — Device Tracking needs to cover throughout production line widely from very early stage of materials through almost final product for consumers. However specific production systems may be somewhat different depending on stages, the following models illustrate typical issues in interrelation between production and tracking.

11.2 *Data around Machine and Circuit Module* — There are many acquired or measured data and context information of such production systems as Manufacturing Execution System (MES) and Work In Process (WIP). Each data or information could be independent or a part of Die Trace Data.

11.2.1 *Obtained Data* — Many sorts of data can be obtained on each piece of equipment. A part of the data or the summarized/processed data could be Die Trace Data.

11.2.1.1 *Examined Data* — If Die Trace Machine is metrology, inspection or test equipment, measured, inspected or tested result data is obtained on the equipment respectively. Such data as a whole, a part of the data or even the data which is summarized/processed, it could be a kind of Die Trace Data. As long as the equipment has one or more modules which have such functions, the equipment can provide the data even if it has process capability.

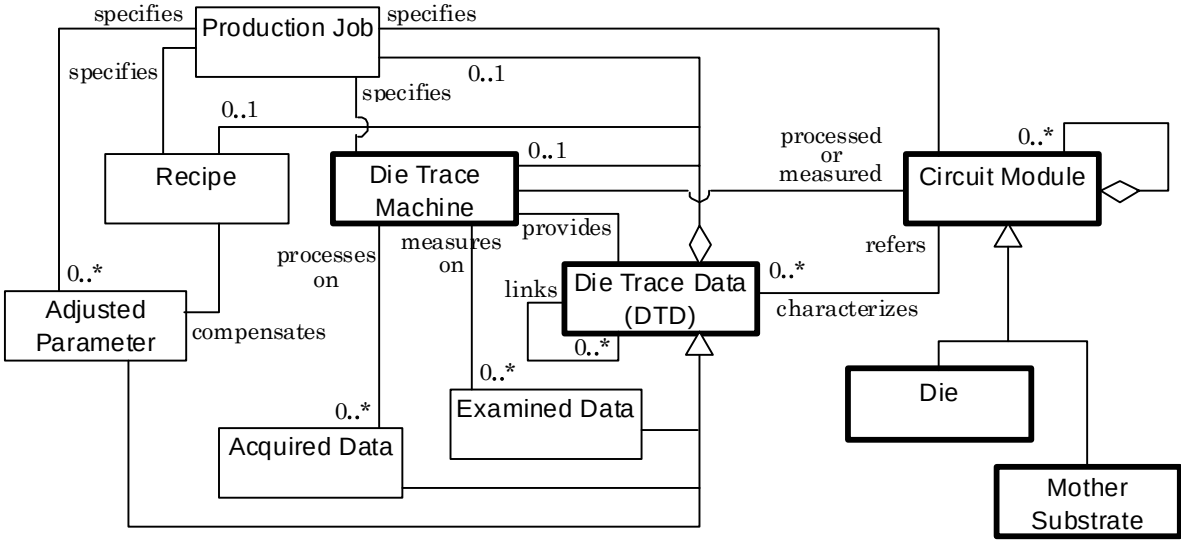

Figure 15
Possible Die Trace Data around Machine and Circuit Module

11.2.1.2 *Acquired Data* — During processing materials on equipment, it may acquire such data as values on sensors and set points on actuators/controllers. Some of the data could be sample values to deduce whether the process is done as expected or not. Acquired data while it is idle, preparation or diagnosis may suggest equipment healthy state. Process logs are the other examples which record detail actions that such instructive prescription as process recipe doesn't describe. In these senses this kind of data could be a kind of Die Trace Data. Identification of mounted/placed device may be one of examples of this data.

11.2.1.3 *Adjusted Parameters* — Process recipes are fixed process prescriptions to instruct or guide how process is done. To avoid increasing number of recipes dramatically, adjustable parameters to allow some flexibility for each recipe is often used. Dispatching/scheduling function of Manufacturing Execution System (MES) or Advanced Process Control (APC) system often assigns their values. These values may help inferring how process was turned out.

11.2.2 *Context Data* — Meaningful data has its property which makes sure where it was obtained, when it was, which phase or step of a course of procedure, how it links with the other data or entities, and so on. Without the property data itself doesn't make sense even if it is so accurate or high resolution. Context data must be a part of Die Trace Data.

11.3 *Die Trace Data to get New Data* — To get Die Trace Data on an equipment for certain process, Die Trace Data for previous process may be required. For example, for devices on a tray to be mounted on lead frames or PCBs, it is difficult to get interrelation between devices and base substrates if identification codes for the devices are unknown while the substrates' identifications are managed. On some limited processes and production lines MES provides a part of Die Trace Data of previous process to equipment. For example, wafer ID information for a carrier is given for 300 mm process equipment. Good die information on a dicing frame is given to die bonding machine because the machine may not be aware of good dice to mount onto lead frames or dice may not tell for inkless systems. However, the information may not be given if dice are provided on such a strip as tray or tape because only good dice are mounted on the strip. Die Tracing system may need to provide such data.

11.3.1 *Quasi Object for Data Transfer* — To fix above problem it should be clarified that Die Trace Machine may need to get a part of Die Trace Data on previous process and Die Tracer may be asked for such data by the machine. To make sure the nature on Die Tracing system, it solves the problem for the system to assume some class object having the nature. Die Trace Staff class on Figure 16 has the role.

11.3.1.1 *Die Trace Staff Quasi Class* — This class has the responsibility to ask some Die Trace Data to Die Tracer if the data is required for tracking. Because it is inherited to Die Trace Machine and Die Tracer, the Machine may ask a part of Die Trace Data to the Tracer and the Tracer can ask the data to the other Die Tracer if the Tracer doesn't have the previous data. This is unofficial class so that this document may not give specific definition. However, the capabilities defined in the other classes provide the its capability.

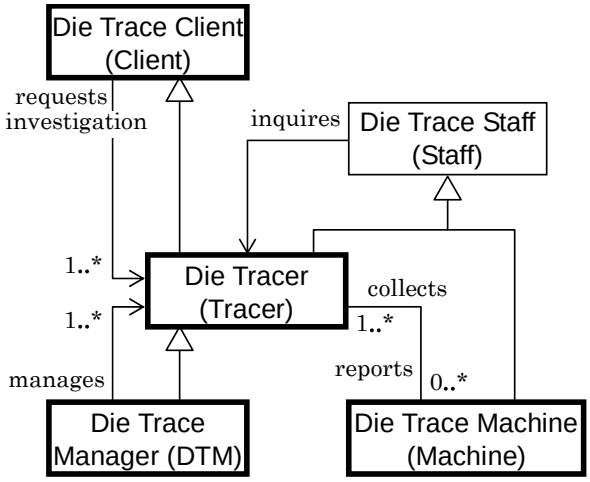

Figure 16
Die Trace Staff Quasi Class

12 Services for Device Tracking

12.1 *Services* — This section defines content of service messages provided by classes in this document.

12.1.1 *Format and Protocol* — The message format and communication protocol may not be defined here because they are dependent to specific implementation.

12.1.2 *Public Interface* — Each service definition is a part of definition of public interface. Implementation of services may not be always required if a set of the message communication between objects or a part of it are not required in intended system or deployment unite provided by user and suppliers.

12.2 *Service Definition* — Service definitions are given for each class.

12.2.1 *CircuitModule* — The following tables defines services of Circuit Module class.

12.2.1.1 *Service List* — Circuit Module class has the following services.

Table 1 Service List for Circuit Module

Operation	*Description*	*Type*	*Reqd*
showProperty	Inquires all or a part of attributes (property) of CircuitModule.	R	Y
showDaughter	Inquires all CircuitModules which this CircuitModule has or mounted on.	R	Y

12.2.1.2 *showProperty* — This service is used to inquire all a part of property of CircuitModule. The following table defines its parameters.

Table 2 Service Definition of showProperty for Circuit Module

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
Designation	C	-	Selected Name in Property.
Property	-	M	Property of CircuitModule.
ServiceStatus	-	M	Result of service request.

12.2.1.3 *showDaughter* — This service is used to inquire CircuitModules mounted on this CircuitModule. The following table defines its parameters.

Table 3 Service Definition of showDaughter for Circuit Module

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
Clarifier	-	M	Clarifier of CircuitModule.
ServiceStatus	-	M	Result of service request.

12.2.1.4 *Parameter List* — Above services use the following parameters.

Table 4 Service Parameter Definitions for Circuit Module

Parameter	*Definition*	*Form*
Clarifier	ID or some code to identify CircuitModule clearly: concatenation of ID and any or all of device type, product name and Manufacturer.	Text.
Designation	One or more attribute names separated by space.	Text.
Property	A part or all of attributes of CircuitModule. However CircuitModule has attributes illustrated on Figure 8 at least, additional attributes are dependent on type and production stage: dice may have Quality or Rank.	Text (structured).
ServiceStatus	Result of service to show successful or reason of failure.	Text.

12.2.2 *DieTraceClient* — The following tables defines services of Die Trace Client class.

12.2.2.1 *Service List* — Die Trace Client class has the following services.

Table 5 Service List for Die Trace Client

Operation	*Description*	*Type*	*Reqd*
showPhenomena	Inquires identification of Attachment.	R	N
setAlert	Inquires name of Attachment.	R	N

12.2.2.2 *showPhenomena* — This service is used to ask posted problems which they have not confirmed or clarified yet. The following table defines its parameters.

Table 6 Service Definition of showPhenomena for Die Trace Client

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
Phenomena	-	C	Phenomena posted on specific device.
ServiceStatus	-	M	Result of service request.

12.2.2.3 *setAlert* — This service is used to notify potential problems which may happen on specific product. The following table defines its parameters.

Table 7 Service Definition of setAlart for Die Trace Client

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
Alert	M	-	Alert which producer has been aware on a product.
ServiceStatus	-	M	Result of service request.

12.2.2.4 *Parameter List* — Above services use the following parameters.

Table 8 Service Parameter Definitions for Die Trace Client

Parameter	*Definition*	*Form*
Alart	Taxonomy ID, Date/Time, Severity, Product, Producer, such boundaries as production period or serial number / lot and description.	Text (structured).
Phenomena	Group Code, Date/Time, Severity, Possibility, Product, Producer, lot or serial numbers, description of problem, User/Site and Application.	Text (structured).
ServiceStatus	Result of service to show successful or reason of failure.	Text.

12.2.3 *DieTraceData* — The following tables defines services of Die Trace Data class.

12.2.3.1 *Service List* — Die Trace Data class has the following services.

Table 9 Service List for Die Trace Data

Operation	*Description*	*Type*	*Reqd*
showProperty	Inquires identification of Attachment.	R	Y
showData	Inquires name of Attachment.	R	Y

12.2.3.2 *showProperty* — This service is used to inquire a part or all of property of Die Trace Data. The following table defines its parameters.

Table 10 Service Definition of showProperty for Die Trace Data

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
Designation	C	-	Selected Name in Property.
Property	-	M	Property of Die Trace Data.
ServiceStatus	-	M	Result of service request.

12.2.3.3 *showData* — This service is used to inquire data value of Die Trace Data. The following table defines its parameters.

Table 11 Service Definition of showData for Die Trace Data

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
DataValue	-	M	Data Value.
ServiceStatus	-	M	Result of service request.

12.2.3.4 *Parameter List* —Above services use the following parameters.

Table 12 Service Parameter Definitions for Die Trace Data

Parameter	*Definition*	*Form*
DataValue	Value of Die Trace Data.	Any form depending on data.
Designation	One or more attribute names separated by space.	Text.
Property	A part or all of attributes of DieTraceData except DataValue.	Text (structured)
ServiceStatus	Result of service to show successful or reason of failure.	Text.

12.2.4 *DieTraceMachine* — The following tables defines services of Die Trace Machine class.

12.2.4.1 *Service List* — Die Trace Machine class has the following services.

Table 13 Service List for Die Trace Machine

Operation	*Description*	*Type*	*Reqd*
acquireDTD	Inquires specific Die Trace Data.	R	Y
setPlanningItems	Sets Die Trace Data items to be reported.	R	N
flush	Releases sending temporary buffered Die Trace Data.	R	N
registerTracer	Changes supervising Die Tracer.	R	N

12.2.4.2 *acquireDTD* — This service is used to acquire Die Trace Data which are available values on Machine or immediate values. The following table defines its parameters.

Table 14 Service Definition of acquireDTD for Die Trace Machine

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
DataName	M	-	Data name to be acquired.
TraceData	-	C	Acquired data value with its property.
ServiceStatus	-	M	Result of service request.

12.2.4.3 *setPlanningItems* — This service is used to set/change planning Die Trace Data to be acquired and reported. The following table defines its parameters.

Table 15 Service Definition of setPlanningItems for Die Trace Machine

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
DataNameList	M	-	Data name list to set.
ServiceStatus	-	M	Result of service request.

12.2.4.4 *flush* — This service is used to release temporary buffered Die Trace Data. Response may be more than once. The following table defines its parameters.

Table 16 Service Definition of flush for Die Trace Machine

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
TraceData	-	C	Buffered available data value with its property.
ServiceStatus	-	M	Result of service request.

12.2.4.5 *registerTracer* — This service is used to set or change supervising Tracer. The following table defines its parameters.

Table 17 Service Definition of registerTracer for Die Trace Machine

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
Tracer	M	-	Representation to specify resisting Tracer.
ServiceStatus	-	M	Result of service request.

12.2.4.6 *Parameter List* — Above services use the following parameters.

Table 18 Service Parameter Definitions for Die Trace Machine

Parameter	*Definition*	*Form*
DataName	DieTraceData name. Suppliers or user shall document data names and their definitions.	Text.
DataNameList	List of Data name separated by space.	Text.
TraceData	Die Trace Data value with its property. See DieTraceDat for the property.	Any form depending on Data.
Tracer	URI or some other code/name to specify DieTracer.	Text.
ServiceStatus	Result of service to show successful or reason of failure.	Text.

12.2.5 *DieTraceManager* — The following tables defines services of Die Trace Manager class.

12.2.5.1 *Service List* — Die Trace Manager class has the following services.

Table 19 Service List for Die Trace Manager

Operation	*Description*	*Type*	*Reqd*
admit	Asks joining Die Tracing network member.	R	Y
resign	Asks retiring from Die tracing network member.	R	Y
devolve	Asks accepting DieTracers as subordinates.	R	N
logException	Asks logging exceptions on a Die Tracer and/or display them.	R	N

12.2.5.2 *admit* — This service is used to join in Die Tracing network to one of its members. The following table defines its parameters.

Table 20 Service Definition of admit for Die Trace Manager

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
ManagingTracer	M	-	Specifier of DieTraceManager to join.
ServiceStatus	-	M	Result of service request.

12.2.5.3 *resign* — This service is used to retiring from Die Tracing network for some reason. The following table defines its parameters.

Table 21 Service Definition of resign for Die Trace Manager

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
ManagingTracer	M	-	Specifier of DieTraceManager to retire.
ServiceStatus	-	M	Result of service request.

12.2.5.4 *devolve* — This service is used to DieTracers from requesting Manager to Serving Manager. The following table defines its parameters.

Table 22 Service Definition of devolve for Die Trace Manager

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
Tracers	M	-	List of specifiers of DieTracers to be transferred.
ServiceStatus	-	M	Result of service request.

12.2.5.5 *logException* — This service is used to transfer exceptions for logging and/or displaying. The following table defines its parameters.

Table 23 Service Definition of logException for Die Trace Manager

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
Exeptions	M	-	List of exceptions to be transferred.
ServiceStatus	-	M	Result of service request.

12.2.5.6 *Parameter List* — Above services use the following parameters.

Table 24 Service Parameter Definitions for Die Trace Manager

Parameter	*Definition*	*Form*
Exceptions	List of exceptions separated by space. Each exception has the following fields: exception ID, classification and description. The last field of each exception has to be terminated with a semicolon (;).	Text (structured).
ManagingTracer	URI or some other code/name to specify DieTraceManager.	Text.
Tracers	List of URI or some other code/name to specify DieTracer, separated by space.	Text.
ServiceStatus	Result of service to show successful or reason of failure.	Text.

12.2.6 *DieTracer* — The following tables defines services of Die Tracer class.

12.2.6.1 *Service List* — Die Tracer class has the following services.

Table 25 Service List for Die Tracer

Operation	*Description*	*Type*	*Reqd*
collectDTD	Asks extra DieTraceData of on some Machine supervised by a DieTracer.	R	N
planDTD	Asks expected DieTraceData on some Machine supervised by a DieTracer.	R	N
showDTD	Inquires specific DieTraceData.	R	Y
setForward	Set forwarding some DieTraceData to specific entity or application.	R	N
catchDTD	Requests to accept forwarded DieTraceData.	R	N
enroll	Asks to be be one of subordinates of DieTraceManager.	R	N
takeDTD	Reports obtained DieTraceData. (This service may be substituted by subscribing events on message broker or middleware and events, depending on implementation.)	N	Y

12.2.6.2 *collectDTD* — This service is used to expect extra DieTraceData to a DieTraceManager. The following table defines its parameters.

Table 26 Service Definition of collectDTD for Die Tracer

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
DataName	M	-	Data name to be acquired.
TraceData	-	C	Collected data value with its property.
ServiceStatus	-	M	Result of service request

12.2.6.3 *planDTD* — This service is used to request expected DieTraceData to DieTraceManager. The following table defines its parameters.

Table 27 Service Definition of planDTD for Die Tracer

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
DataNameList	M	-	Data names to be collected.
Target	M	-	Possible data source or group.
ServiceStatus	-	M	Result of service request

12.2.6.4 *showDTD* — This service is used to inquire specific DieTraceData. The following table defines its parameters.

Table 28 Service Definition of showDTD for Die Tracer

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
DataName	M	-	Data name to see.
Target	C	-	Possible data source or group.
TraceData	-	C	Obtained data value with its property.
ServiceStatus	-	M	Result of service request

12.2.6.5 *setForward* — This service is used to forward specific DieTraceData to entity or application when the data are acquired. The following table defines its parameters.

Table 29 Service Definition of setForward for Die Tracer

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
Destination	M	-	Destination of forwarded data.
DataNameList	M	-	Data name list to be forwarded.
Target	C	-	Possible data source or group.
ServiceStatus	-	M	Result of service request

12.2.6.6 *catchDTD* — This service is used to transfer specific DieTracedata from the other Tracer. The following table defines its parameters.

Table 30 Service Definition of catchDTD for Die Tracer

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
TraceData	M	-	Data value with its property.
ServiceStatus	-	M	Result of service request

12.2.6.7 *enroll* — This service is used to ask under control of requester DieTraceManager for device tracking. The following table defines its parameters.

Table 31 Service Definition of enroll for Die Tracer

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
ManagingTracer	M	-	Specifier of DieTraceManager.
ServiceStatus	-	M	Result of service request

12.2.6.8 *takeDTD* — This service is used to report obtained DieTraceData. The following table defines its parameters.

Table 32 Service Definition of takeDTD for Die Tracer

Parameter	*Req/Ind*	*Rsp/Conf*	*Description*
TraceData	M	-	Data value with its property.
ServiceStatus	-	M	Result of service request

12.2.6.9 *Parameter List* — Above services use the following parameters.

Table 33 Service Parameter Definitions for Die Tracer

Parameter	*Definition*	*Form*
DataName	DieTraceData name. Suppliers or user shall document data names and their definitions.	Text.
DataNameList	List of Data name separated by space.	Text.
Destination	URI or some other code/name to specify forwarded application. If it is null, it means removal of existing forwarding setups for specified data.	Text
ManagingTracer	URI or some other code/name to specify DieTraceManager.	Text
Target	Code/name to specify process or machine.	Text
TraceData	Die Trace Data value with its property. See DieTraceDat for the property.	Any form depending on Data.
ServiceStatus	Result of service to show successful or reason of failure.	Text

APPENDIX 1
APPENDIX Device Tracking

NOTICE: The material in this appendix is an official part of SEMI T13 and was approved by full letter ballot procedures on April 30, 2004.

A1-1 Device Tracking

A1-1.1 *Die Tracking vs. Device Tracking*

A1-1.1.1 Growing modern semiconductor technology allows more than one die in a package. Such package technologies as Flip Chip Packaging and System In Package (SIP) are examples. Sometimes more than five dice are packaged in a small plastic package. Because the word “device” is the final product of semiconductor, tracking devices can be covered by tracking die specification.

A1-1.2 *Device Tracking Data* — Devices are moved from one substrate to another. The dimensions of the substrates are not the same so inevitably there will not be a one-to-one relationship between the “From” substrate and the “To” substrate. The simplest solution would be to send a set of reports each of which include the “From” substrate type, ID and XY devices coordinates and the same type of information again for the “To” substrate. This will, however, lead to a lot of repetition of substrate type and ID.

A1-1.2.1 *Tracking Report* — In order to keep it compact, the report will contain an ordered array of elements for each device on the “To” substrate. The “From” substrate type and Id are only provided when substrate changes. The array may be row major or column major to match the processing sequence used on equipment, otherwise the “From” substrate type and Id might change back and forth.

A1-1.2.2 *Device Tracking Example* — An example of Device Tracking is provided in Related Information.

A1-1.3 *Device ID Data* — Devices may be marked either while they are still on the wafer (ref: Backside WaferId TF) or when they are packaged on a strip, or in a tray.

A1-1.3.1 *Device ID Reporting Example* — An example of Device ID Reporting is provided in Related Information..

A1-1.4 *Substrate Layout Data* — Device Tracking Data and Device ID Data depend on the devices to which they refer being arranged in a regular two dimensional array or matrix. I some cases, for example a multi-chip module, this is not the case. The Substrate Layout Data contains a representation of the substrate geometry in terms of repeating and nested matrices. Each matrix works as an identifier which links those elements to the corresponding Device Tracking Data and Device ID Data.

A1-1.4.1 *Substrate Layout Data Example* — example of Substrate Layout Data Reporting is provided in Related Information.

APPENDIX 2
COORDINATE SYSTEMS

NOTICE: The material in this appendix is an official part of SEMI T13 and was approved by full letter ballot procedures on April 30, 2004.

A2-1 Horizontal Coordinates

A2-1.1 *Die Coordinate, Origin and Fiducial* — Definition of the origin of coordinates used with this specification is basically compliant to such SEMI standard as G81 and G84. For complicated or irregular die arrangement including sub-array of dice, it is assumed to be compliant to Section 18 Coordinate System in SEMI E130. If the other coordinate system or origin definitions, which are specific to substrates or user/equipment dependent, user or supplier shall specify in specification or complementary document. Fiducial information shall be also documented if it is necessary for tracking.

A2-1.2 *The other Coordinate Systems* — If surface position is required to give Die Trace Data except Device coordinates, such SEMI standard as M17, M20 or M21 shall be basically used: e.g. for visual inspection and film thickness measurement. If some derivations from the standards or complete different coordinate system are required, depending on equipment, supplier or user shall document to exchange specification.

A2-2 Vertical Dimension

A2-2.1 *Vertical Position* — Sometimes vertical position may be required to represent Die Trace Data. This standard assumes one of the following representation of vertical position.

A2-2.1.1 *Z Dimension* — This representation consists of ‘Order’ and ‘Height’. ‘Order’ means order number of concerning device from lowest substrate. ‘Height’ is actual vertical dimension of concerning device from top surface of the device just below it or substrate. Unit of the Height is given with the value or common with the other information. See R1-3 for example.

A2-2.1.2 *GCD Grid* — When above positioning system is not convenient or doesn’t work for specific equipment or tracking system, this system is used alternatively. Take Greatest Common Denominator (GCD) of all mounted devices on substrate and assume a grid of the GCD. It may be 50 micron for example. The origin of the grid must be surface of base substrate and minus vale may be used in such a case that a die is implanted in the substrate. This may be useful for some advanced assembly or bird-eye view preferred systems.

A2-2.2 *The Other Vertical dimensions* — Additional vertical dimension may be required for some equipment. Because they are equipment dependent, this document doesn’t specify anything about coordinates except the previous subsections. User or suppliers shall document to understand Die Trace Data correctly.

RELATED INFORMATION 1
DEVICE TRACKING EXAMPLES

NOTICE: This related information is not an official part of SEMI T13 and was derived from the Japanese Traceability Committee. This related information was approved for publication by full letter ballot on April 30, 2004.

R1-1 Device Tracking Report

R1-1.1 The following figurer and list are example of Device Tracking Report.

R1-1.1.1 *Device Example* — The following Figure shows a device picked from a wafer and placed on a strip.

R1-1.1.2 *Report Example* — The following list shows an example of the report for the first 2 devices on a strip taken from Wafer1 and the next 2 taken from Wafer2.

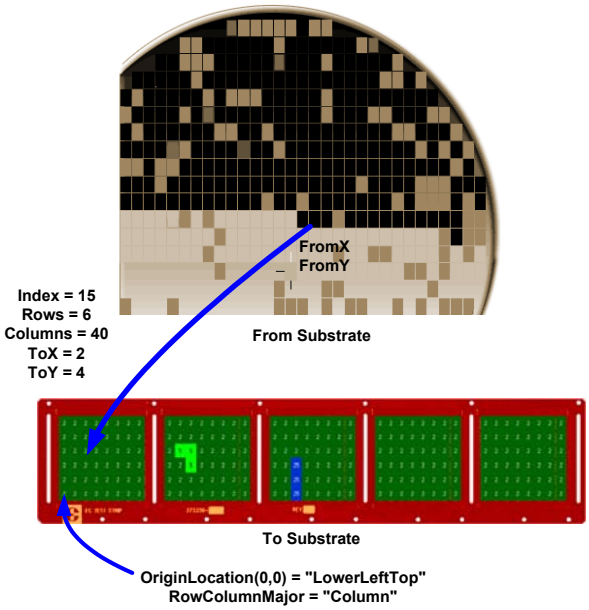

Figure R1-1

List 1 An Example of Device Tracking Data in XML

```
<?xml version="1.0" encoding="utf-8" ?>
<DeviceTrackingData xmlns="urn:semi-org:xsd.Doc3754.V0312.DeviceTrackingData">
      <SubstrateType>Strip</SubstrateType>
      <SubstrateId>MyStrip</SubstrateId>
      <LayoutId>Device</LayoutId>
      <OriginLocation>LowerLeftTop</OriginLocation>
      <RowColumnMajor>Column</RowColumnMajor>
      <FromSubstrate>
            <SubstrateType>Wafer</SubstrateType>
            <SubstrateId>Wafer1</SubstrateId>
            <D><X>10</X><Y>18</Y></D>
            <D><X>7</X><Y>17</Y></D>
      </FromSubstrate>
      <FromSubstrate>
            <SubstrateType>Wafer</SubstrateType>
            <SubstrateId>Wafer2</SubstrateId>
            <D><X>18</X><Y>1</Y></D>
            <D><X>9</X><Y>0</Y></D>
      </FromSubstrate>
</DeviceTrackingData>
```

NOTE 1: The <D> elements contain the X and Y location on the substrate from which they came.

NOTE 2: The <FromSubstrate> element specifies the substrate type and identifier from which the devices came from.

NOTE 3: The <D> elements represent each device on the layout “Device” for strip “MyStrip”.

NOTE 4: The number of <D> entries must equal Rows * Columns defined in the “Device” layout.

NOTE 5: The one dimensional array of <D> elements represents the two dimensional array of locations defined in the “Devices” layout in RowColumnMajor format with respect to the OriginLocation.

NOTE 6: The OriginLocation corresponds to the definition of the same name in SEMI G81.

R1-2 Device ID Report

R1-2.1 *ID Example* — The following figure shows a strip of devices with alphanumeric and 2D matrix marks. The unique part of the 2D mark is “LOT001-XXYY”.

R1-2.2 *Report Example* — The following list shows a truncated version of the Device Id Data report for the strip shown in upper figure.

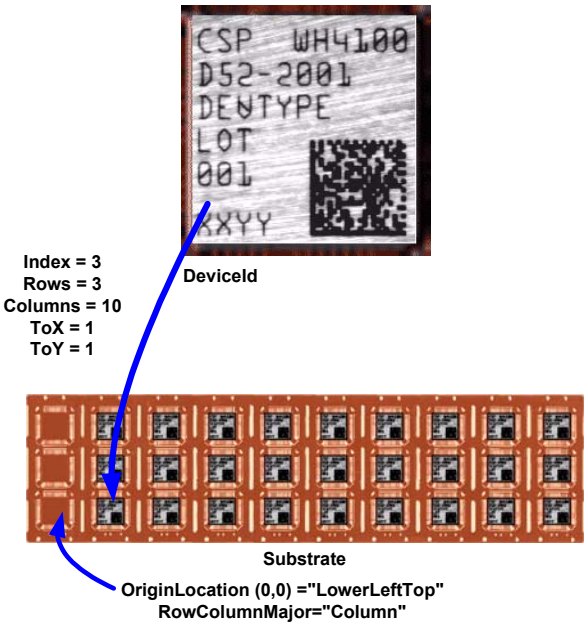

Figure R1-2

List 2 An Example of Device ID Data in XML

```
<?xml version="1.0" encoding="utf-8" ?>
<DeviceIdData xmlns="urn:semi-org:xsd.Doc3754.V0312.DeviceIdData">
      <SubstrateType>Strip</SubstrateType>
      <SubstrateId>MyStrip</SubstrateId>
      <LayoutId>Device</LayoutId>
      <OriginLocation>LowerLeftTop</OriginLocation>
      <RowColumnMajor>Column</RowColumnMajor>
      <D><Id></Id></D>
      <D><Id></Id></D>
      <D><Id></Id></D>
      <D><Id>Lot001-XXYY</Id></D>
      <D><Id>etc.</Id></D>
</DeviceIdData>
```

NOTE 1:The <D> elements contain a tracking code marked on the package that is unique for the manufacturer and product.

NOTE 2: The <D> elements represent each device on the layout “Device” for strip “MyStrip”.

NOTE 3: The number of <D> entries must equal Rows * Columns defined in the “Device” layout.

NOTE 4: The one dimensional array of <D> elements represents the two dimensional array of locations defined in the “Devices” layout in RowColumnMajor format with respect to the OriginLocation.

NOTE 5: The OriginLocation corresponds to the definition of the same name in SEMI G81.

R1-3 Substrate Layout Data Report

R1-3.1 *Substrate Layout Example* — The following figure shows the lower left corner of a strip that contains an array of SRAM devices with 2 Flash devices mounted on each.

R1-3.2 *Layout Representation Example* — The following list shows the substrate layout data that represents the strip shown in the figure.

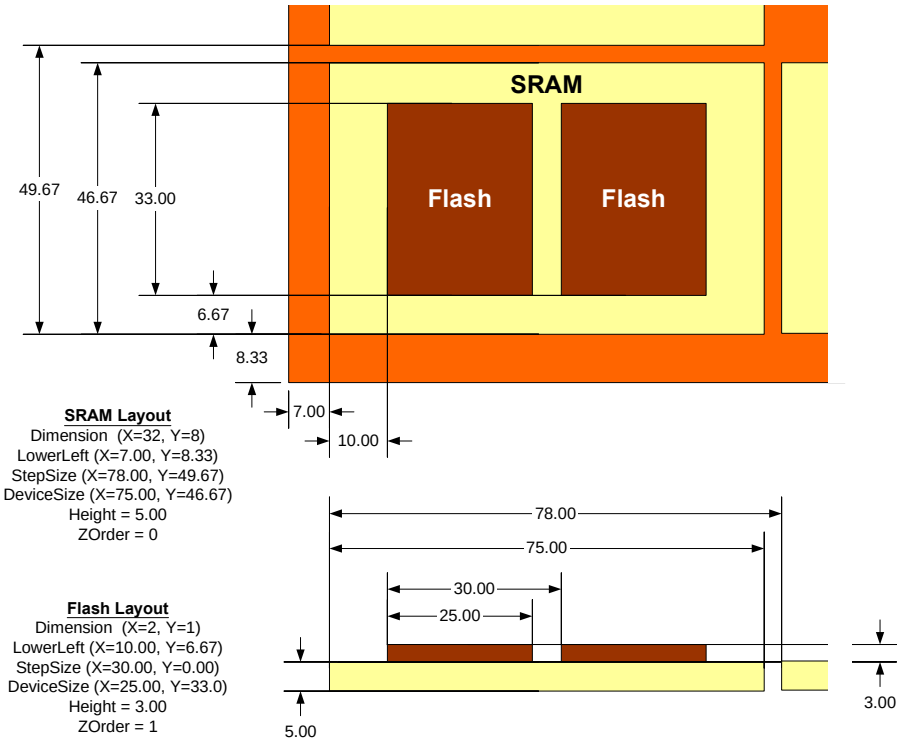

Figure R1-3

List 3 An Example of Substrate Layout Data in XML

```
<?xml version="1.0" encoding="utf-8" ?>
<SubstrateLayoutData xmlns="urn:semi-
org:xsd.Doc3754.V0312.SubstrateLayoutData">
      <SubstrateType>Strip</SubstrateType>
      <SubstrateId>MyStrip</SubstrateId>
      <LayoutId>Device</LayoutId>
      <Units>100micron</Units>
      <Size><X>2510.00</X><Y>414.02</Y></Size>
      <Layout>
            <LayoutId>SRAM</LayoutId>
            <Dimension><X>32</X><Y>8</Y></Dimension>
            <LowerLeft><X>7.00</X><Y>8.33</Y></LowerLeft>
            <StepSize><X>78.00</X><Y>49.67</Y></StepSize>
            <DeviceSize><X>75.00</X><Y>46.67</Y></DeviceSize>
            <Z><Order>0</Order><Height>5.00</Height></Z>
            <Layout>
                  <LayoutId>FLASH</LayoutId>
                  <Dimension><X>2</X><Y>1</Y></Dimension>
                  <LowerLeft><X>10.00</X><Y>6.67</Y></LowerLeft>
                  <StepSize><X>30.00</X><Y>0.00</Y></StepSize>
                  <DeviceSize><X>25.00</X><Y>33.00</Y></DeviceSize>
                  <Z><Order>1</Order><Height>3.00</Height></Z>
            </Layout>
      </Layout>
</SubstrateLayoutData>
```

RELATED INFORMATION 2
AN EXAMPLE OF ENVELOP AND HEADERS FOR XML COMMUNICATION

NOTICE: This related information is not an official part of SEMI T13 and was derived from the Japanese Traceability Committee. This related information was approved for publication by full letter ballot on April 30, 2004.

R2-1 Message Encapsulation in Envelopes

R2-1.1 *Message Envelope* — Communicating messages for Die Trace Data in this document with eXtensible Markup Language (XML) is one of effective choice. Such XML communication protocols as Simple Object Access Protocol (SOAP) and XML-RPC (Remote Procedure Call) are very popular in this industry. In such protocols, messages are structured and encapsulated in such lower level protocol as Hyper Text Transfer Protocol (HTTP). The encapsulation is referred to as Envelope. Actual message header and body are filled in the envelop. This related information introduces some example of the envelopes in the following sections.

R2-2 Placing Message Name in Header with Message Parameters in Body

R2-2.1 *Regular SOAP or XML-RPC Messaging* — If message has no possibility to have binary value as its parameter, regular XML messaging is often used. It is typical to have message name in header and parameters in body as shown in the following list.

R2-2.1.1 *SEMI E128 Style Data Report* — The following list shows an example of regular XML messaging envelope in accordance with SEMI E128 “Specification for XML Message Structure” for the service ‘takeDTD’ of Die Tracer. Service name is enclosed in element ‘Action’ of SOAP Header and message parameters are enclosed in element ‘data’ of SOAP Body. SOAP Fault may be appended in the SOAP Body if SOAP communication error happens.

List 4 XML Messaging Envelope Example 1

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV=
"http://schemas.xmlsoap.org/soap/envelope/">
  <SOAP-ENV:Header>
     <MessageHeader xmlns="urn:semi-org:schema:xmlmsg:E128:0703"
          elementFormDefault="qualified">
        <From>MdmAbc001</From>
        <To>DtHalNykA012</To>
        <MessageType>REQUEST</MessageType>
        <RequestId>123456</RequestId>
        <Action>takeDTD</Action>
        <ReplyExpected>true</ReplyExpected>
     </MessageHeader>
  </SOAP-ENV:Header>
  <SOAP-ENV:Body>
     <data>
        <DeviceIdData xmlns="urn:semi-org:xsd.Doc3754.V0312.DeviceIdData">
           <SubstrateType>Strip</SubstrateType>
           <SubstrateId>MyStrip</SubstrateId>
           <LayoutId>Device</LayoutId>
           <OriginLocation>LowerLeftTop</OriginLocation>
           <RowColumnMajor>Column</RowColumnMajor>
           <D><Id></Id></D>
           <D><Id></Id></D>
           <D><Id></Id></D>
           <D><Id>Lot001-XXYY</Id></D>
           <D><Id>etc.</Id></D>
        </DeviceIdData>
     </data>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

NOTE 1: The above list is just an example and some details may not be fully compliant to SEMI E128 or related documents.

R2-3 Placing Message Name and Parameters in the Second MIME Block

R2-3.1 *Messaging with Attachment* — If message may have binary value in its parameter, an attachment may be required. If transferred data is binary format, some contrivance to encode the data is necessary unless the data is transferred separately by another means: e.g. binary file transfer. This case XML Message has an attached block for the data.

R2-3.1.1 *IPC-2501 Style Data Report* — The following list shows an example of XML messaging envelope with attachment in accordance with IPC 2501 “Definition of Web-Based Exchange of XML Data (Message Broker)” for the service ‘takeDTD’ of Die Tracer. SOAP Envelope in the first MIME Block has just message’s information in element ‘MessageInfo’ of SOAP Header and SOAP Fault in SOAP Body if it is required. Message name and message parameters are encapsulated in the second MIME Block.

List 5 XML Messaging Envelope Example 2

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAP-ENV:Envelope xmlns:SOAP-
ENV="http://schemas.xmlsoap.org/soap/envelope/"
 elementFormDefault="qualified">
 <SOAP-ENV:Header>
    <MessageInfo xmlns:IPC2501MI=
     "http://webstds.gatech.edu/2501/MessageInfo.xsd"
     elementFormDefault="qualified"
     dateTime="2003-03-27T15:54:17.02-05:00"
     sender="MdmAbc001"
     destination="DtHalNykA012"
     messageId="123456"
     messageSchema="takeDTD">
    </MessageInfo>
 </SOAP-ENV:Header>
 <SOAP-ENV:Body>
 </SOAP-ENV:Body>
</SOAP-ENV:Envelope>

--- Second MIME Block
<?xml version="1.0" encoding="UTF-8"?>
     <takeDTD xmlns:TxxxDtTD=
       "urn:semi-org:TC-Schema:Txxx:0704:DTO-TDS"
       elementFormDefault="qualified"
       dateTime="2003-03-27T15:54:17.02-05:00">
       <DeviceIdData xmlns="urn:semi-org:xsd.Doc3754.V0312.DeviceIdData">
          <SubstrateType>Strip</SubstrateType>
          <SubstrateId>MyStrip</SubstrateId>
          <LayoutId>Device</LayoutId>
          <OriginLocation>LowerLeftTop</OriginLocation>
          <RowColumnMajor>Column</RowColumnMajor>
          <D><Id></Id></D>
          <D><Id></Id></D>
          <D><Id></Id></D>
          <D><Id>Lot001-XXYY</Id></D>
          <D><Id>etc.</Id></D>
       </DeviceIdData>
     </takeDTD>
```

NOTE 2: Above list is just an example and some details may not be fully compliant to IPC-2501 or related documents.

RELATED INFORMATION 3
AN EXAMPLE OF TRANSACTIONS HAPPENS IN DIE TRACING SYSTEM

NOTICE: This related information is not an official part of SEMI T13 and was derived from the Japanese Traceability Committee. This related information was approved for publication by full letter ballot on April 30, 2004.

R3-1 An Example of Transactions for Device Tracking

R3-1.1 The following sequence diagram shows typical transaction in Die/Device Tracking System. This diagram illustrate an application how tracking runs and an example how this specification is applied.

R3-1.2 *Characters in the Example* — This example has a couple of die tracing system islands. Each of them is supervised by independent Die Trace Manager: Mf and Mn. They may be separated areas in a frontend fab, frontend and backend fabs, or semiconductor fab and SMT fab. Each island has three Die Tracers including the Manager. Each Tracer has a couple of Die Trace Machines which can provide traceability functions and they must be doing effective process for tracking. A Die Trace Client is located close to later process system: Cx.

R3-1.3 *Script* — The example shows transactions related to a certain products. Transactions flow through the following stages.

R3-1.3.1 *Usual Data Report* — Each Machine report Die Trace Data for specific material for the product on individual process.

R3-1.3.2 *Initial Inquiry* — The client needs to look for suspicious process to screen possible devices which may have problem. It asks certain Die Trace Data to find the evidence of the problem to its one of familiar Tracers.

R3-1.3.3 *Secondary Inquiry* — The first Tracer Tk doesn't have the data, so it asks its familiar Tracer or its Manager. Unfortunately the Manager Mn/Ti doesn't have that, so it asks to its subordinate Tracers: this case Tj only.

R3-1.3.4 *Extended Inquiry* — Because no subordinates have the data, there is nothing for the Manager but to inquire the other Managers on tracking network. Inquired Manager Mf/Tc looks for to itself and asks its subordinate Tracers.

R3-1.3.5 *Replying Back* — Fortunately one of the Tracers has the data and it responds to the Manager. The Manager transfers to the other Manager as the response. The Client ended up with expected data.

R3-1.3.6 *Flushed Inquiry* — Above inquiries may be flushed at one time rather than step by step inquiry.

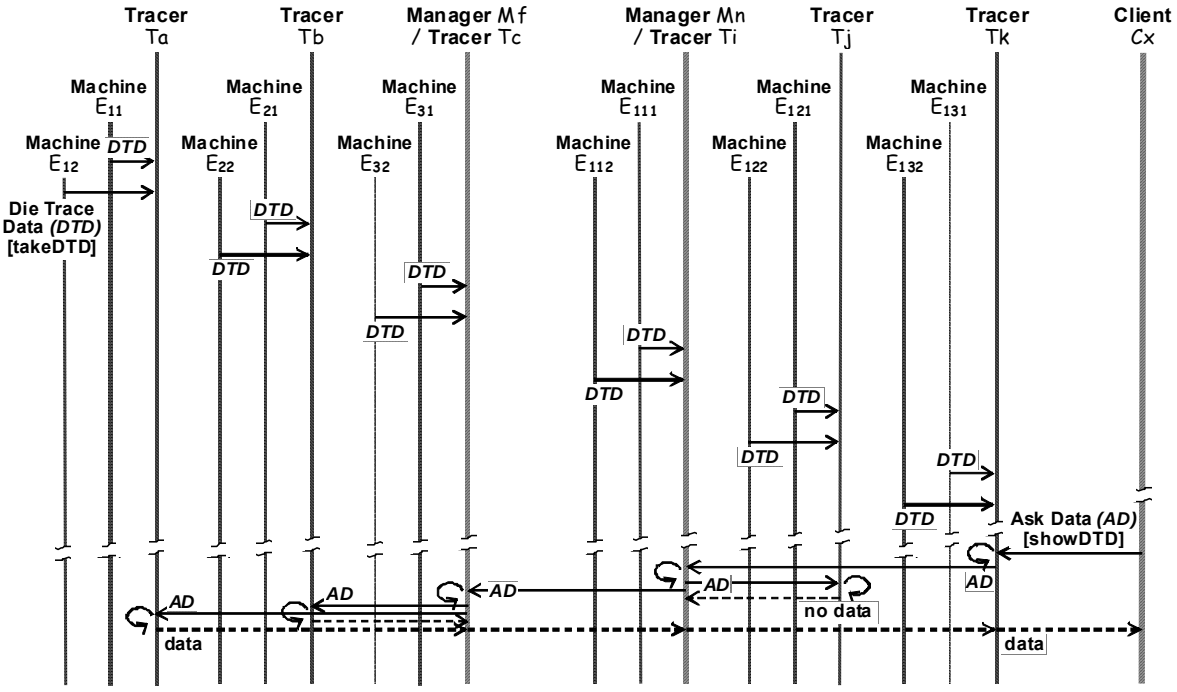

Figure R3-1
An Example Transaction

RELATED INFORMATION 4
USE CASE OF DEVICE TRACKING

NOTICE: This related information is not an official part of SEMI T13 and was derived from the Japanese Traceability Committee. This related information was approved for publication by full letter ballot on April 30, 2004.

R4-1 Use Case Diagram

R4-1.1 Use Case diagram is one of well-known means for understanding and analysis of problem domain to start with. It is often used before modeling a system or bringing a good solution.

R4-1.2 Use Case diagram consists of Actors and Use Cases. A use case defines something expected to a system related to an actor or between actors. All use cases build up whole system.

R4-2 Use Case Diagram of Device Tracking

R4-2.1 The following diagram shows use case for Device Tracking.

R4-2.2 *Trace* — Consumer asks listing all or a part of stages of Production History and Information for tracking. Examples may be picking up five latest history items or histories for changing substrates. Product Information may include original production country, test house identification, assembly subcontractor and so on.

R4-2.3 *Back Track* — Consumer traces Production History from the latest one and checks also Problem Histories of the state. If no specific essential problems is not found on the stage, information for previous stage is provided interactively. If something is unclear, further information or data is requested to Producer.

R4-2.4 *Update History* — Producer append / updates Production History describing what has done on the product including changing location or substrate. Information attached to the history may be dependent to the stage, equipment and/or product. Producer may also append / update Problem History describing briefly what the problem was and how it was fixed.

R4-2.5 *Check History* — Consumer asks detail history of a stage. System replies all or expected categories of Production History and Information. Because sometimes Consumer's curiosity may beyond what system knows, the system may ask further information to responsible producer in such case.

R4-2.6 *Notify* — Consumer notifies phenomena on a product to system when some problem happens on the product. The system looks for resemble phenomena registered on the system. If it identified, the system let the consumer know all information about the phenomena and update registry. Otherwise the system registers the phenomena and asks all producers or most possible ones to fix it. When a responsible producer responds about the problem, the system update registry and let the consumer know.

R4-2.7 *Alert* — Producer issues potential problems for a product. This information is registered on system and distributed to all possible consumers. This capability can be used even if the product is still in process on some stage.

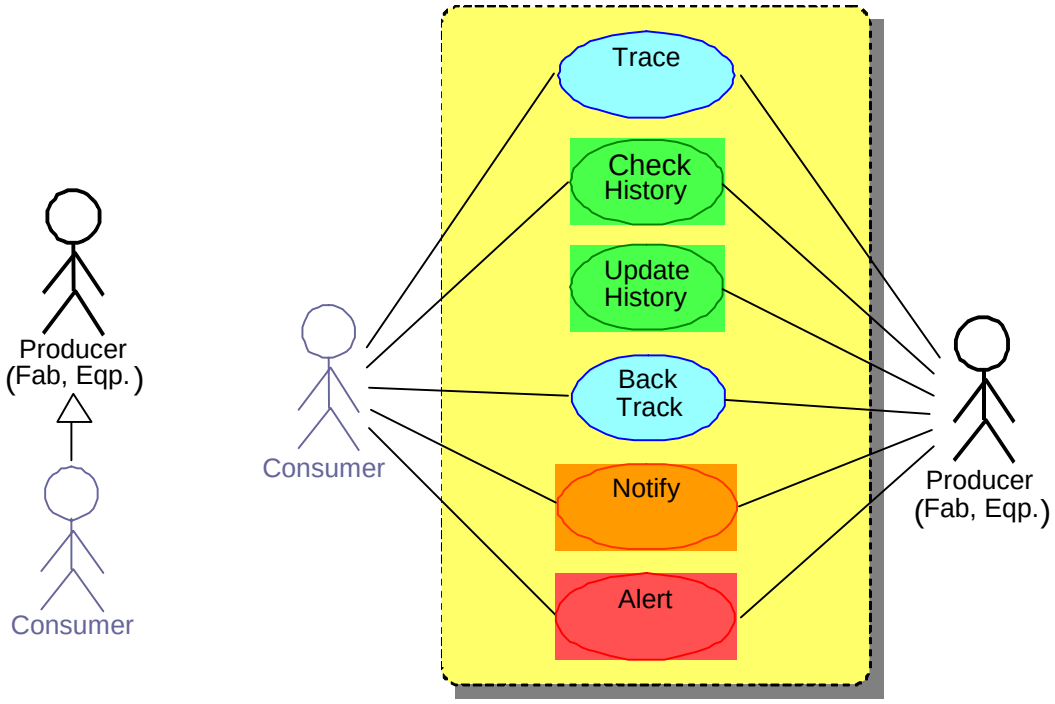

Figure R4-1
Use Case Diagram for Device Tracking

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

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

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

SEMI T13.1-1104
SPECIFICATION FOR SECS PROTOCOL FOR DEVICE TRACKING

This specification was technically approved by the Global Traceability Committee and is the direct responsibility of the Japanese Traceability Committee. Current edition approved by the Japanese Regional Standards Committee on July 23, 2004. Initially available at www.semi.org September 2004; to be published November 2004.

1 Purpose

1.1 The purpose of this document is to map services and data in SEMI T13 into SECS protocol.

2 Scope

2.1 This document is applied to SECS implementation of semiconductor device traceability with some interoperability with semiconductor Manufacturing Execution System (MES).

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

3 Referenced Standards

3.1 *SEMI Standards*

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

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

SEMI G84 — Specification for Strip Map Protocol

SEMI T12 — Specification for Tracing Jigs and Implements

SEMI T13 — Specification for Device Tracking: Concepts, Behavior and Services

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

4 Requirements

4.1 *Service Mapping* — This specification maps service instructions defined in SEMI T13 into SECS stream/functions as follows. Supporting Class names specified in the following table are used as OBJSPEC or a part of OBJSPEC to request services.

Table 1 Service Message Instruction Mapping

Service Message Name	*Supporting Class*	*Stream, Function*	*SECS-II Message Name*
acquireDTD	DieTraceMachine	S14, F19/F20	Generic Service Request/Response
admit	DieTraceManager	S14, F19/F20	Generic Service Request/Response
catchDTD	DieTracer	S14, F19/F20	Generic Service Request/Response
collectDTD	DieTracer	S14, F19/F20	Generic Service Request/Response
devolve	DieTraceManager	S14, F19/F20	Generic Service Request/Response
enroll	DieTracer	S14, F19/F20	Generic Service Request/Response
flush	DieTraceMachine	S14, F19/F20	Generic Service Request/Response
logException	DieTraceManager	S14, F19/F20	Generic Service Request/Response
planDTD	DieTracer	S14, F19/F20	Generic Service Request/Response
registerTracer	DieTraceMachine	S14, F19/F20	Generic Service Request/Response
resign	DieTraceManager	S14, F19/F20	Generic Service Request/Response
setAlert	DieTraceClient	S14, F19/F20	Generic Service Request/Response
setForward	DieTracer	S14, F19/F20	Generic Service Request/Response
setPlanningItems	DieTraceMachine	S14, F19/F20	Generic Service Request/Response

SEMI T13.1-1104 © SEMI 2004

Service Message Name	*Supporting Class*	*Stream, Function*	*SECS-II Message Name*
showData	DieTraceData	S14, F19/F20	Generic Service Request/Response
showDaughter	CircuitModule	S14, F19/F20	Generic Service Request/Response
showDTD	DieTracer	S14, F19/F20	Generic Service Request/Response
showPhenomena	DieTraceClient	S14, F19/F20	Generic Service Request/Response
showProperty	CircuitModule, DieTraceData	S14, F19/F20	Generic Service Request/Response
takeDTD	DieTracer	S14, F19/F20	Generic Service Request/Response

4.2 *Service Parameter Mapping* — The following table shows the mapping between service parameters defined by SEMI T13 and data items defined by SEMI E5. The data item SVCNAME used with S14F19 is the text string defined in the above table. The data item SPNAME used with S14F19/F20 is also text string defined in following table. It is recommended but not limited to be case sensitive for both data items for future extension.

Table 2 Service Message Parameter Mapping

Parameter Name	*SECS-II Data Item Reference*	*SECS-II Format*	*Remarks*
Alert	L, 2 1. <SPNAME> 2. <SPVAL>	SPNAME = Alert SPVAL = Item (00)	SPVAL: n-element list of property item.
Clarifier	L, 2 1. <SPNAME> 2. <SPVAL>	SPNAME = Clarifier SPVAL = (20)	
DataName	L, 2 1. <SPNAME> 2. <SPVAL>	SPNAME = DataName SPVAL = (20)	
DataNameList	L, 2 1. <SPNAME> 2. <SPVAL>	SPNAME = DataNameList SPVAL = (20)	
DataValue	L, 2 1. <SPNAME> 2. <SPVAL>	SPNAME = DataValue SPVAL = (00, 1x, 2x, 3x, 4x, 5x)	"x" is an appropriate octal defined in SEMI E5.
Designation	L, 2 1. <SPNAME> 2. <SPVAL>	SPNAME = Designation SPVAL = (20)	If SPVAL is null or just spaces, it means all. Spaces could be separator.
Exceptions	L, 2 1. <SPNAME> 2. <SPVAL>	SPNAME = Exceptions SPVAL = Item (00)	SPVAL: n-element list of property item.
MachineType	L, 2 1. <SPNAME> 2. <SPVAL>	SPNAME = MachineType SPVAL = (20)	
ManagingTracer	L, 2 1. <SPNAME> 2. <SPVAL>	SPNAME = ManagingTracer SPVAL = (20)	
Phenomena	L, 2 1. <SPNAME> 2. <SPVAL>	SPNAME = Phenomena SPVAL = Item (00)	SPVAL: n-element list of property item.
Property	L, 2 1. <SPNAME> 2. <SPVAL>	SPNAME = Property SPVAL = Item (00)	SPVAL: n-element list of property item.

Parameter Name	*SECS-II Data Item Reference*	*SECS-II Format*	*Remarks*
ServiceStatus	L, 2 1. <SVCACK> 2. L, p 1. L, 2 1. <ERRCODE$_1$> 2. <ERRTEXT$_1$> ….. ….. ….. p. L, 2 1. <ERRCODE$_p$> 2. <ERRTEXT$_p$>	SVCACK = (10) ERRCODE = (51, 52, 54) ERRTEXT = (20)	If service was successful, P = 0 and no error list is attached. If not, the maximum length of each ERRTEXT must be 80.
Target	L, 2 1. <SPNAME> 2. <SPVAL>	SPNAME = Target SPVAL = (20)	
TraceData	L, 2 1. <SPNAME> 2. <SPVAL>	SPNAME = TraceData SPVAL = (00, 1x, 2x, 3x, 4x, 5x)	"x" is an appropriate octal defined in SEMI E5.
Tracer	L, 2 1. <SPNAME> 2. <SPVAL>	SPNAME = Tracer SPVAL = (20)	
Tracers	L, 2 1. <SPNAME> 2. <SPVAL>	SPNAME = Tracers SPVAL = (20)	
Item	L, 2 1. <SPNAME> 2. <SPVAL>	PNAME = (20) PVAL = (00, 20)	
Name of Service	<SVCNAME>	20	*Note:* message name other than message parameter.

4.3 *The Other Communication* — The other SECS-II communication may additionally take place to run manufacturing systems. Such communications as event report and exception notification are usually done from equipment to host. Also some instructive communications for manufacturing as job management and remote command are done from host to equipment. But they are just recommended to do regularly and independently to this specification because they are not in the scope of traceability but in production scope.

RELATED INFORMATION 1
SECS MAPPING FOR RESTRICTED INFORMATION

NOTICE: This related information is not an official part of SEMI T13.1 and was derived from the Japanese Traceability Committee. This related information was approved for publication by full letter ballot on July 23, 2004.

R1-1 Particular Mapping for Specific Equipment

R1-1.1 Often some backend equipment may just handle such characteristic data of each device on a strip substrate as sorting or classification data and transfer information. Characteristic information exchanging service messages and their parameters are mapped as shown in the following tables. This mapping is one of possible specific mappings for arrayed characteristic device information on a strip and there may not be a full mapping but just for particular functions related to the exchanging services. This mapping is based on SEMI G84 with some extension.

Table R1-1 Service Instructions Mapping

Service Message Name	*Supporting Class*	*Stream, Function*	*SECS-II Message Name*
showData	DieTraceData	S14, F1/F2	Get Attribute Request / Data.
ShowProperty	DieTraceData	S14, F1/F2	Get Attribute Request / Data.
takeDTD	DieTracer	S6, F11/F12	Selected Equipment Status Request / Data.

Table R1-2 Service Parameters Mapping

Parameter Name	*SECS-II Data Item Reference*	*SECS-II Format*	*Remarks*
DataValue	L, 2 1. <ATTRID> 2. <ATTRDATA>	ATTRID = (20) ATTRDATA = (0, 1x, 2x, 3x, 4x, 5x)	showData of DieTraceData (specific use for data content, or whole characteristic map data on a substrate including some properties).
Designation	<ATTRID>	ATTRID = (20)	showProperty of DieTraceData.
Property	L, 2 1. <ATTRID> 2. <ATTRDATA>	ATTRID = (20) ATTRDATA = (0, 1x, 2x, 3x, 4x, 5x)	showProperty of DieTraceData.
ServiceStatus	<ACKC6>	ACKC6=(10)	showData, showProperty.
	<OBJACK>	OBJACK=(51)	takeDTD of DieTracer.
TraceData	<V>	V=(0, 1x, 2x, 3x, 4x, 5x)	takeDTD of DieTracer.

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

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

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

SEMI T13.2-1104
SPECIFICATION FOR XML PROTOCOL FOR DEVICE TRACKING

This specification was technically approved by the Global Traceability Committee and is the direct responsibility of the Japanese Traceability Committee. Current edition approved by the Japanese Regional Standards Committee on July 23, 2004. Initially available at www.semi.org September 2004; to be published November 2004.

1 Purpose

1.1 The purpose of this document is to map Services and data in SEMI T13 into XML protocol.

2 Scope

2.1 This document is applied to XML implementation of Jigs and Implements traceability with some interoperability with Surface Mount Technology (SMT) equipment.

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

3 Limitations

3.1 This protocol mapping has a consideration being interoperable with SMT manufacturing it may be used in such area, using this specification on such manufacturing area is not guaranteed because this specification is originally planned to be working on semiconductor manufacturing systems including equipment.

4 Referenced Standards

4.1 *SEMI Standards*

SEMI T12 — Specification for Tracing Jigs and Implements

SEMI T13 — Specification of Device Tracking: Concepts, Behavior and Services

4.2 *Association Connecting Electronics Industries (IPC)/National Electronics Manufacturing Initiative (NEMI)/American National Standard institute (ANSI)*[1]

IPC-2501 — Definition for Web-Based Exchange of XML data; (July, 2003)

IPC-2541 — Generic Requirements for Electronics Manufacturing Shop-Floor Equipment Communication (CAMX); (October, 2001 / ANSI Approved November, 2001)

4.3 *World Wide Web Consortium (W3C)*[2]

REC-xml-20001006 — Extensible Markup Language (XML) 1.0 (Second Edition)

NOTE-SOAP-20000508 — Simple Object Access Protocol (SOAP) 1.1

4.4 *Internet Engineering Task Force*[3]

RFC2045 — Multipurpose Internet Message Extensions (MIME) Part 1: Format of Internet Message Bodies

RFC2616 — Hyper Text Transfer Protocol – HTTP/1.1

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

5 Terminology

5.1 *Abbreviations & Acronyms*

5.1.1 *MIME* — Multipurpose Internet Message Extensions

5.1.2 *SOAP* — Simple Object Access Protocol

5.1.3 *XML* — eXtensible Markup Language

6 Base Requirements

6.1 *Base Protocol Mapping* — This XML protocol specification adopts Simple Object Access Protocol (SOAP) over Hyper Text Transfer Protocol (HTTP) on Transmission Control Protocol/Internet Protocol (TCP/IP). Some equipment users produce not only semiconductor devices but also Print Circuit Boards (PCBs). It is convenient for such users to have common messaging protocol with standard messaging for Surface Mount Technology (SMT). For this purpose this protocol mapping extension standard specifies the messaging mechanism conformant to IPC-2501 “Definition for Web-Based Exchange of XML Data” as

1 http://www.ipc.org, http://www.nemi.org, http://www.ansi.org, The original developer is IPC with NEMI cooperation, ANSI ends up with authorizing approval.

2 World Wide Web Consortium, 32 Vassar St. Room 32-G515, Cambridg, MA 02139, USA Telephone: 617.253.2613; Fax: 617.258.5999, http://www.w3c.org,

3 http://www.ietf.org, IETF Secretariat c/o Corporation for National Research Initiatives, 1895 Preston White Drive, Suite 100, Reston, VA 20191-5434, USA, Telephone +1 703 620 8990, Fax +1 703 620 9071

one of mapping means. Also this extension standard document follows IPC-2541 “Generic Requirements for Electronics Manufacturing Shop-Floor Equipment Communication Messages” and IPC-2551 “Sectional Requirements for Manufacturing Execution System” for that reason. The following subsections describe a part of these specifications for introduction. Refer to these documents for detail reference.

6.2 *SOAP Structure* — A HTTP message in this specification has an envelope to contain a couple of MIME blocks. One is for SOAP Envelope and the other is for message detail to be able to mix with such non-XML information as binary data. The service message specified in SEMI T12 is contained in latter one. SOAP standard envelope MIME Block contains SOAP Header which has message information and SOAP Body which contains SOAP Faults. Outlined diagram is shown in Related Information. Official definitions are given in referenced documents.

6.3 *Messaging Mechanism* — Message Broker, that is logical middle ware server to handle and relay messages, is assumed somewhere on communication network. While the broker behaves as communication server, the other logical nodes are message clients. Every message starts with a client and responded by the server. Messages communicated between clients are exchanged through the server. The server keeps posted messages from clients in a queue for each individual client. Expecting recipient client asks the server for messages and a topmost message in the queue for the client is responded if the queue is not empty. The recipient client posts acknowledge back to the server. When the recipient replies back a message to the original sender client, same thing happens with switched roles between sender and recipient clients. Explanatory diagrams for this outline are shown in Related Information. Official specification is given in referenced documents.

6.4 *Issues Specific to this Messaging* — A few issues for IPC based communication protocol for Jigs and Implements Tracking are introduced here.

6.4.1 *Domain Configuration* — Message Broker has domain configuration information to make sure all acting entities, possible messages with related publisher, recipient and subscriber clients. Also it has other detail characteristics of the domain of communication. Every client needs to look up the information before the first messaging.

6.4.2 *Message Information* — This is an element contained in SOAP Header. This information provides time of transaction, sender, destination, message identification and schema type used for service message detail in the second MIME block.

6.4.3 *Attributes in Service Message Element* — The top element of each service message has date/time of the message, session ID or session reference, request ID or request reference, and optional values. Some of the optional values are extended for Jigs and Implements Tracking communication: linkId, linkCnt and linkExp. The extension may be used for delayed reply message and multiple divided replies to keep conformance of related messages.

7 Requirements

7.1 *Service Mapping* — Services defined in SEMI T12 are mapped on the following table.

7.1.1 *Capitalized Service Message Element Name Base* — Each service message is mapped into an XML element with capitalized heading letter to prevent collision with XML or XML Schema native words.

7.1.2 *Service Message Element Name Suffix* — Because of symmetry of message broker communication and for XML element naming convention, suffix of ‘Request’ or ‘Response’ is added to each service message element name for request or response respectively: e.g. AddEventRequest to request service and AddEventResponse for its response message.

Table 1 Service Instruction Mapping

Service Message Name	*Supporting Class*	*Service Message Element Name Base*	*Remarks*
acquireDTD	DieTraceMachine	AcquireDTD	
admit	DieTraceManager	Admit	
catchDTD	DieTracer	CatchDTD	
collectDTD	DieTracer	CollectDTD	
devolve	DieTraceManager	Devolve	
enroll	DieTracer	Enroll	
flush	DieTraceMachine	Flush	

Service Message Name	*Supporting Class*	*Service Message Element Name Base*	*Remarks*
logException	DieTraceManager	LogException	
planDTD	DieTracer	PlanDTD	
registerTracer	DieTraceMachine	RegisterTracer	
resign	DieTraceManager	Resign	
setAlert	DieTraceClient	SetAlert	
setForward	DieTracer	SetForward	
setPlanningItems	DieTraceMachine	SetPlanningItems	
showData	DieTraceData	ShowData	
showDaughter	CircuitModule	ShowDaughter	
showDTD	DieTracer	ShowDTD	
showPhenomena	DieTraceClient	ShowPhenomena	
showProperty	CircuitModule, DieTraceData	ShowProperty	
takeDTD	DieTracer	TakeDTD	

7.2 *Service Parameter Mapping* — Service parameter type mapping is shown in the following table.

Table 2 Service Parameter Type Mapping

Parameter Name	*Attribute/Element*	*Type*	*Remarks*
Alert	Element	anyType	Formatted Text
Clarifier	Element	string	
DataName	Element	string	
DataNameList	Element	string	Repeating DataName
DataValue	Element	anyType	Formatted Text
Designation	Element	string	
Exceptions	Element	anyType	Formatted Text
MachineType	Element	string	
ManagingTracer	Element	string	
Phenomena	Element	anyType	Formatted Text
Property	Element	anyType	Formatted Text
ServiceStatus	Element	complexType	status+Rejection
Target	Element	string	
TraceData	Element	anyType	Formatted Text
Tracer	Element	string	
Tracers	Element	string	Repeating Tracer

7.3 *Additional Parameters for Service Messages* — Because this protocol mapping works for asynchronous HTTP server-client communication on message broker, such additional parameters to give complementary information as issued time and transaction identification are required to be attached to each message. The information may be dependent to the role of the message. Type mapping of them are shown in the following table.

Table 3 Type Mapping of Additional Parameters

Parameter Name	*Attribute/Element*	*Type*	*Remarks*
dateTime	Attribute	dateTime	
sessionId	Attribute	string	
sessionRef	Attribute	string	
requestId	Attribute	string	
requestRef	Attribute	string	
linkId	Attribute	string	extended for delayed response

Parameter Name	*Attribute/Element*	*Type*	*Remarks*
linkCnt	Attribute	integer	extended for delayed response
linkExp	Attribute	duration	extended for delayed response

7.4 *Example XML Message Fragments* — The following subsections give examples of some fragments of service messages in XML. A full message is given with envelops and headers.

7.4.1 *showProperty Request* — The following XML description is an example of the principal part of service request message for showProperty provided by CircuitModule objects. Because the Designation of parameters for a CircuitModule is dependent to the circuit module and tracing objectives, they should be discussed between trace data provider and consumer.

```
<ShowPropertyRequest dateTime=”2004-03-29T13:05:25.000-08:00”
 SessionRef=”SemiDtQs-9876543210” RequestId=”123456”>
  <Designation>Id</Designation>
  <Designation>Type</Designation>
  <Designation>ProductName</Designation>
  <Designation>Lot</Designation>
  </History>
</ShowPropertyRequest>
```

7.4.2 *showPhenomena Response* — The following XML description is an example of the principal part of service response message for showPhenomena provided by DieTraceClient objects. Detail information of argument “Phenomena” is not specified in the standard because it’s dependent to target product. Additional definition should be exchanged in the system.

```
<ShowPhenomenaResponse dateTime=”2004-03-29T09:07:59.000-08:00”
 SessionRef=”SemiDtQs-9876543123” RequestRef=”123443” status=”GRANTED”>
  <Phenomena>
    <Registered>2004-03-26T18:07:51.000-08:00<Registered>
    <Product>meson689-4A</Product>
    <Problem>Noise sensitive</Problem>
    <Problem>Restarted Occasionally</Problem>
    <Application>Office Computers</Application>
  </Phenomena>
  <ServiceStatus>
    <SvcAck>Successful</SvcAck>
  </ServiceStatus>
</ShowPhenomenaResponse>
```

APPENDIX 1
APPENDIX TITLE

NOTICE: The material in this appendix is an official part of SEMI T13.2 and was approved by full letter ballot procedures on July 23, 2004.

A1-1 XML Message Schema

A1-1.1 *XML Schema* — XML Schema is a kind of template for documents. The following XML Schemas shows how messages in this XML protocol are represented.

A1-1.1.1 *Message Schema* — Each message in T12 has one XML Schema. Parsing with the schema, application can understand what elements and attributes mean. Also it can check syntax easily.

A1-1.1.2 *Common Definitions* — Common part of the schema is separated and it is included in each message schema.

A1-1.2 *Enumeration* — Attachment specific or eqiopment specific enumeration may not be addressed here. Corresponding part of the shema should be extended or restricted with ‘redefine’ element, for example, in application schema.

```
XML Message Schema Name:  T13ShowDaughterRequestCmo.xsd      for CircuitModule Class

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - - -  Show Daughter Request - - - - - - - - - - - - - - -  -->
<xsd:element name="ShowDaughterRequest" type="ShowDaughterRequestType"/>

<xsd:complexType name="ShowDaughterRequestType">
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>


XML Message Schema Name:  T13ShowDaughterResponseCmo.xsd      for Circuit Module Class

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - -  - - - -     Show Daughter Response - - - -  - - - -    -->
<xsd:element name="ShowDaughterResponse" type="ShowDaughterResponseType"/>

<xsd:complexType name="ShowDaughterResponseType">
  <xsd:sequence>
    <xsd:element name="Clarifier" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13ShowPropertyRequestCmo.xsd for CircuitModule Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - -  - - - -     Show Property Request - - - -  - - - -    -->
<xsd:element name="ShowPropertyRequest" type="ShowPropertyRequestType"/>

<xsd:complexType name="ShowPropertyRequestType">
  <xsd:sequence>
    <xsd:element name="Designation" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13ShowPropertyResponseCmo.xsd for CircuitModule Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - -  - - - -     Show Property Response - - - -  - - - -    -->
<xsd:element name="ShowPropertyResponse" type="ShowPropertyResponseType"/>

<xsd:complexType name="ShowPropertyResponseType">
  <xsd:sequence>
    <xsd:element name="Property" type="xsd:anyType" minOccurs="0"
     maxOccurs="unbounded"/>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13ShowPhenomenaRequestDtc.xsd for DieTraceClient Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - -  - - - -     Show Phenomena Request - - - -  - - - -    -->
<xsd:element name="ShowPhenomenaRequest" type="ShowPhenomenaRequestType"/>
```

```
<xsd:complexType name="ShowPhenomenaRequestType">
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13ShowPhenomenaResponseDtc.xsd for DieTraceClient Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - -  - - - -     Show Phenomena Response - - - -  - - - -    -->
<xsd:element name="ShowPhenomenaResponse" type="ShowPhenomenaResponseType"/>

<xsd:complexType name="ShowPhenomenaResponseType">
  <xsd:sequence>
    <xsd:element name="Phenomena" type="xsd:anyType" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13SetAlertRequestDtc.xsd for DieTraceClient Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - -  - - - -     Set Alert Request - - - -  - - - -    -->
<xsd:element name="SetAlertRequest" type="SetAlertRequestType"/>

<xsd:complexType name="SetAlertRequestType">
  <xsd:sequence>
    <xsd:element name="Alert" type="xsd:anyType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13SetAlertResponseDtc.xsd for DieTraceClient Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">
```

```
<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - -  - - - -     Set Alert Response - - - -  - - - -    -->
<xsd:element name="SetAlertResponse" type="SetAlertResponseType"/>

<xsd:complexType name="SetAlertResponseType">
  <xsd:sequence>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13ShowDataRequestDtd.xsd for DieTraceData Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - -  - - - -     Show Data Request - - - -  - - - -    -->
<xsd:element name="ShowDataRequest" type="ShowDataRequestType"/>

<xsd:complexType name="ShowDataRequestType">
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13ShowDataResponseDtd.xsd for DieTraceData Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - -  - - - -     Show Data Response - - - -  - - - -    -->
<xsd:element name="ShowDataResponse" type="ShowDataResponseType"/>

<xsd:complexType name="ShowDataResponseType">
  <xsd:sequence>
    <xsd:element name="DataValue" type="xsd:anyType"/>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13ShowPropertyRequestDtd.xsd for DieTraceData Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
```

```
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - -  - - - -     Show Property Request - - - -  - - - -    -->
<xsd:element name="ShowPropertyRequest" type="ShowPropertyRequestType"/>

<xsd:complexType name="ShowPropertyRequestType">
  <xsd:sequence>
    <xsd:element name="Designation" type="xsd:string" minOccurs="0"
maxOccurs="unbounded"/>>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13ShowPropertyResponseDtd.xsd for DieTraceData Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - -  - - - -     Show Property Response - - - -  - - - -    -->
<xsd:element name="ShowPropertyResponse" type="ShowPropertyResponseType"/>

<xsd:complexType name="ShowPropertyResponseType">
  <xsd:sequence>
    <xsd:element name="Property" type="xsd:anyType" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13AcquireDTDRequestDtmc.xsd for DieTraceMachine Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - -  - - - -     Acquire DTD Request - - - -  - - - -    -->
<xsd:element name="AcquireDTDRequest" type="AcquireDTDRequestType"/>

<xsd:complexType name="AcquireDTDRequestType">
  <xsd:sequence>
    <xsd:element name="DataName" type="xsd:string"/>
  </xsd:sequence>
```

```
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13AcquireDTDResponseDtmc.xsd for DieTraceMachine Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - -  - - - -     Acquire DTD Response - - - -  - - - -    -->
<xsd:element name="AcquireDTDResponse" type="AcquireDTDResponseType"/>

<xsd:complexType name="AcquireDTDResponseType">
  <xsd:sequence>
    <xsd:element name="TraceData" type="xsd:anyType" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13SetPlanningItemsRequestDtmc.xsd for DieTraceMachine Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - -  - - - -     Set Planning Items Request - - - -  - - - -    -->
<xsd:element name="SetPlanningItemsRequest" type="SetPlanningItemsRequestType"/>

<xsd:complexType name="SetPlanningItemsRequestType">
  <xsd:sequence>
    <xsd:element name="DataName" type="xsd:string" minOccurs="1"
maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13SetPlanningItemsResponseDtmc.xsd for DieTraceMachine Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>
```

```
<!-- - - - -  - - - -     Set Planning Items Response - - - -  - - - -    -->
<xsd:element name="SetPlanningItemsResponse" type="SetPlanningItemsResponseType"/>

<xsd:complexType name="SetPlanningItemsResponseType">
  <xsd:sequence>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13FlushRequestDtmc.xsd for DieTraceMachine Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - -  - - - -     Flush Request - - - -  - - - -    -->
<xsd:element name="FlushRequest" type="FlushRequestType"/>

<xsd:complexType name="FlushRequestType">
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13FlushResponseDtmc.xsd for DieTraceMachine Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - -  - - - -     Flush Response - - - -  - - - -    -->
<xsd:element name="FlushResponse" type="FlushResponseType"/>

<xsd:complexType name="FlushResponseType">
  <xsd:sequence>
    <xsd:element name="DieTraceData" type="xsd:anyType" minOccurs="0"
maxOccurs="unbounded"/>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13RegisterTracerRequestDtmc.xsd for DieTraceMachine Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
```

```
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - -  - - - -     Register Tracer Request - - - -  - - - -    -->
<xsd:element name="RegisterTracerRequest" type="RegisterTracerRequestType"/>

<xsd:complexType name="RegisterTracerRequestType">
  <xsd:sequence>
    <xsd:element name="Tracer" type="xsd:string"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13RegisterTracerResponseDtmc.xsd for DieTraceMachine Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - -  - - - -     Register Tracer Response - - - -  - - - -    -->
<xsd:element name="RegisterTracerResponse" type="RegisterTracerResponseType"/>

<xsd:complexType name="RegisterTracerResponseType">
  <xsd:sequence>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13AdmitRequestDtmg.xsd for DieTraceManager Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - -  - - - -     Admit Request - - - -  - - - -    -->
<xsd:element name="AdmitRequest" type="AdmitRequestType"/>

<xsd:complexType name="AdmitRequestType">
  <xsd:sequence>
    <xsd:element name="ManagingTracer" type="xsd:string"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13AdmitResponseDtmg.xsd for DieTraceManager Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - - - - -     Admit Response - - - -  - - - -    -->
<xsd:element name="AdmitResponse" type="AdmitResponseType"/>

<xsd:complexType name="AdmitResponseType">
  <xsd:sequence>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13ResignRequestDtmg.xsd for DieTraceManager Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!--  - - - - - - Resign Request - - - - - -     -->
<xsd:element name="ResignRequest" type="ResignRequestType"/>

<xsd:complexType name="ResignRequestType">
  <xsd:sequence>
    <xsd:element name="ManagingTracer" type="xsd:string"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13SResignResponseDtmg.xsd for DieTraceManager Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - - - -   Resign Response - - - - - -  - - -->
<xsd:element name="ResignResponse" type="ResignResponseType"/>

<xsd:complexType name="ResignResponseType">
  <xsd:sequence>
```

```
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13DevolveRequestDtmg.xsd for DieTraceManager Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - - -   Devolve Request - - - - - -  - - -->
<xsd:element name="DevolveRequest" type="DevolveRequestType"/>

<xsd:complexType name="DevolveRequestType">
  <xsd:sequence>
    <xsd:element name="Tracer" type="xsd:string" minOccurs="1" maxOccurs="unbounded"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13DevolveResponseDtmg.xsd for DieTraceManager Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - - -   Devolve Response - - - - - -  - - -->
<xsd:element name="DevolveResponse" type="DevolveResponseType"/>

<xsd:complexType name="DevolveResponseType">
  <xsd:sequence>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13LogExceptionRequestDtmg.xsd for DieTraceManager Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>
```

```
<!-- - - - - - - -   Log Exception Request - - - - - -  - - -->
<xsd:element name="LogExceptionRequest" type="LogExceptionRequestType"/>

<xsd:complexType name="LogExceptionRequestType">
  <xsd:sequence>
    <xsd:element name="Exceptions" type="xsd:anyType" minOccurs=”1”
maxOccurs=”unbounded”/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13LogExceptionResponseDtmg.xsd   for DieTraceManager Class

```
<xsd:schema xmlns:xsd=”http://www.w3.org/2001/XMLSchema”
 xmlns:jit=”http://www.semi.org/Traceability/T13.2-V01”
 targetNamespace=”http://www.semi.org/Traceability/T13.2-V01”
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - - -   Log Exception Response - - - - - -  - - -->
<xsd:element name="LogExceptionResponse" type="LogExceptionResponseType"/>

<xsd:complexType name="LogExceptionResponseType">
  <xsd:sequence>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13CollectDTDRequestDtr.xsd   for DieTracer Class

```
<xsd:schema xmlns:xsd=”http://www.w3.org/2001/XMLSchema”
 xmlns:jit=”http://www.semi.org/Traceability/T13.2-V01”
 targetNamespace=”http://www.semi.org/Traceability/T13.2-V01”
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - - -   Collect DTD Request - - - - - -  - - -->
<xsd:element name="CollectDTDRequest" type="CollectDTDRequestType"/>

<xsd:complexType name="CollectDTDRequestType">
  <xsd:sequence>
    <xsd:element name="DataName" type="xsd:string"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13CollectDTDResponseDtr.xsd   for DieTracer Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - - -   Collect DTD Response - - - - - -  - - -->
<xsd:element name="CollectDTDResponse" type="CollectDTDResponseType"/>

<xsd:complexType name="CollectDTDResponseType">
  <xsd:sequence>
    <xsd:element name="TraceData" type="xsd:anyType" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13PlanDTDRequestDtr.xsd for DieTracer Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - - -   Plan DTD Request - - - - - -  - - -->
<xsd:element name="PlanDTDRequest" type="PlanDTDRequestType"/>

<xsd:complexType name="PlanDTDRequestType">
  <xsd:sequence>
    <xsd:element name="DataName" type="xsd:string" minOccurs="1"
maxOccurs="unbounded"/>
    <xsd:element name="Target" type="xsd:string"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13PlanDTDResponseDtr.xsd for DieTracer Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - - -   Plan DTD Response - - - - - -  - - -->
<xsd:element name="PlanDTDResponse" type="PlanDTDResponseType"/>

<xsd:complexType name="PlanDTDResponseType">
  <xsd:sequence>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
```

```
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13ShowDTDRequestDtr.xsd for ExpiryDieTracer Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - - -   Show DTD Request - - - - - -  - - -->
<xsd:element name="ShowDTDRequest" type="ShowDTDRequestType"/>

<xsd:complexType name="ShowDTDRequestType">
  <xsd:sequence>
    <xsd:element name="DataName" type="xsd:string"/>
    <xsd:element name="Target" type="xsd:string" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13ShowDTDResponseDtr.xsd for DieTracer Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - - -   Show DTD Response - - - - - -  - - -->
<xsd:element name="ShowDTDResponse" type="ShowDTDResponseType"/>

<xsd:complexType name="ShowDTDResponseType">
  <xsd:sequence>
    <xsd:element name="DieTraceData" type="xsd:anyType" minOccurs="0" maxOccurs="1"/>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13SetForwardRequestDtr.xsd for DieTracer Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
```

```
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - - - - - Set Forward Request - - - - - - - - - - - - -->
<xsd:element name="SetForwardRequest" type="SetForwardRequestType"/>

<xsd:complexType name="SetForwardRequestType">
  <xsd:sequence>
    <xsd:element name="Destination" type="xsd:string"/>
    <xsd:element name="DataName" type="xsd:string" minOccurs="1"
maxOccurs="unbounded"/>
    <xsd:element name="Target" type="xsd:string" minOccurs="0" maxOccurs="1"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>


XML Message Schema Name:  T13SetForwardResponseDtr.xsd      for DieTracer Class

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - - - - - Set Forward Response - - - - - - - - - - - - -->
<xsd:element name="SetForwardResponse" type="SetForwardResponseType"/>

<xsd:complexType name="SetForwardResponseType">
  <xsd:sequence>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>


XML Message Schema Name:  T13CatchDTDRequestDtr.xsd      for DieTracer Class

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - - -   Catch DTD Request - - - - - -  - - -->
<xsd:element name="CatchDTDRequest" type="CatchDTDRequestType"/>

<xsd:complexType name="CatchDTDRequestType">
  <xsd:sequence>
    <xsd:element name="DieTraceData" type="xsd:anyType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13CatchDTDResponseDtr.xsd   for DieTracer Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - - -   Catch DTD Response - - - - - -  - - -->
<xsd:element name="CatchDTDResponse" type="CatchDTDResponseType"/>

<xsd:complexType name="CatchDTDResponseType">
  <xsd:sequence>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13EnrollRequestDtr.xsd   for DieTracer Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - - -   Enroll Request - - - - - -  - - -->
<xsd:element name="EnrollRequest" type="EnrollRequestType"/>

<xsd:complexType name="EnrollRequestType">
  <xsd:sequence>
    <xsd:element name="ManagingTracer" type="xsd:string"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13EnrollResponseDtr.xsd   for DieTracer Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - - -   Enroll Response - - - - - -  - - -->
<xsd:element name="EnrollResponse" type="EnrollResponseType"/>

<xsd:complexType name="EnrollResponseType">
  <xsd:sequence>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
```

```
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13TakeDTDRequestDtr.xsd for DieTracer Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - - -   Take DTD Request - - - - - -  - - -->
<xsd:element name="TakeDTDRequest" type="TakeDTDRequestType"/>

<xsd:complexType name="TakeDTDRequestType">
  <xsd:sequence>
    <xsd:element name="DieTraceData" type="xsd:anyType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegRequestAttr"/>
</xsd:complexType>

</xsd:schema>
```

XML Message Schema Name: T13TakeDTDResponseDtr.xsd for DieTracer Class

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http://www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<xsd:include
 schemaLocation="http://www.semi.org/Traceability/T13.2-V01/CommonDefinitions"/>

<!-- - - - - - - -   Take DTD Response - - - - - -  - - -->
<xsd:element name="TakeDTDResponse" type="TakeDTDResponseType"/>

<xsd:complexType name="TakeDTDResponseType">
  <xsd:sequence>
    <xsd:element name="ServiceStatus" type="ServiceStatusType"/>
  </xsd:sequence>
  <xsd:attributeGroup ref="RegResponseAttr"/>
</xsd:complexType>

</xsd:schema>
```

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
 xmlns:jit="http:// www.semi.org/Traceability/T13.2-V01"
 targetNamespace="http://www.semi.org/Traceability/T13.2-V01"
 elementFormDefault="qualified" attributeFormDefault="qualified">

<!-- ==========###### Global Definitions ################ - - -->
<!-- Genaric Types - - - -  Generic Types - - - -  Generic Types - - -->
```