Interface A (EDA) Standards

Fan-Tien Cheng

Institute of Manufacturing Engineering

National Cheng Kung University

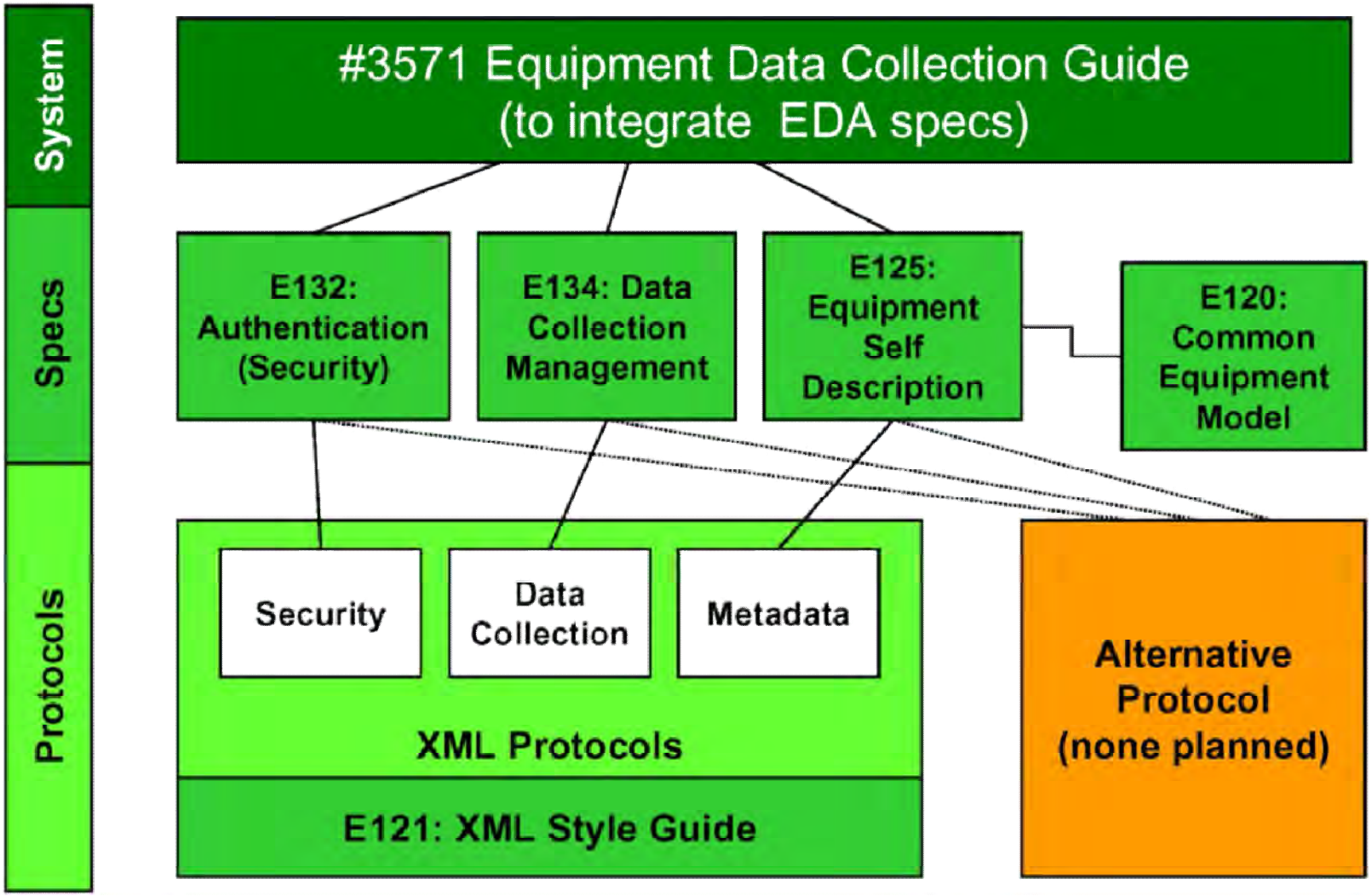
July 31, 2005

Agenda

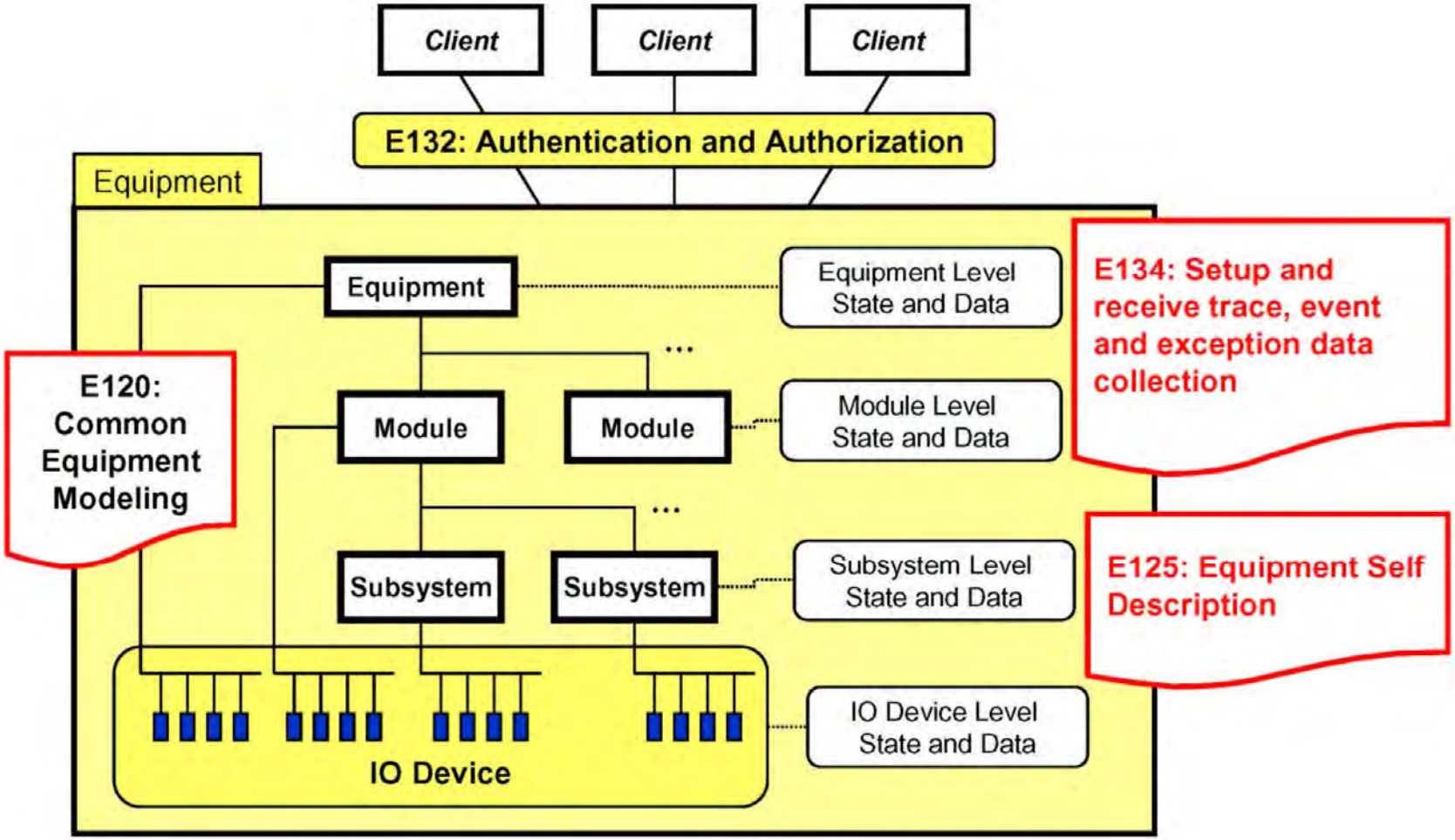
- **EDA Overview**
- **Equipment Client Authentication and Authorization (E132)**
- **Common Equipment Model (E120)**
- **Equipment Self Description (E125)**
- **Data Collection Management (E134)**

P.S.: Portion of this introduction material is edited from the training material of ISMI.

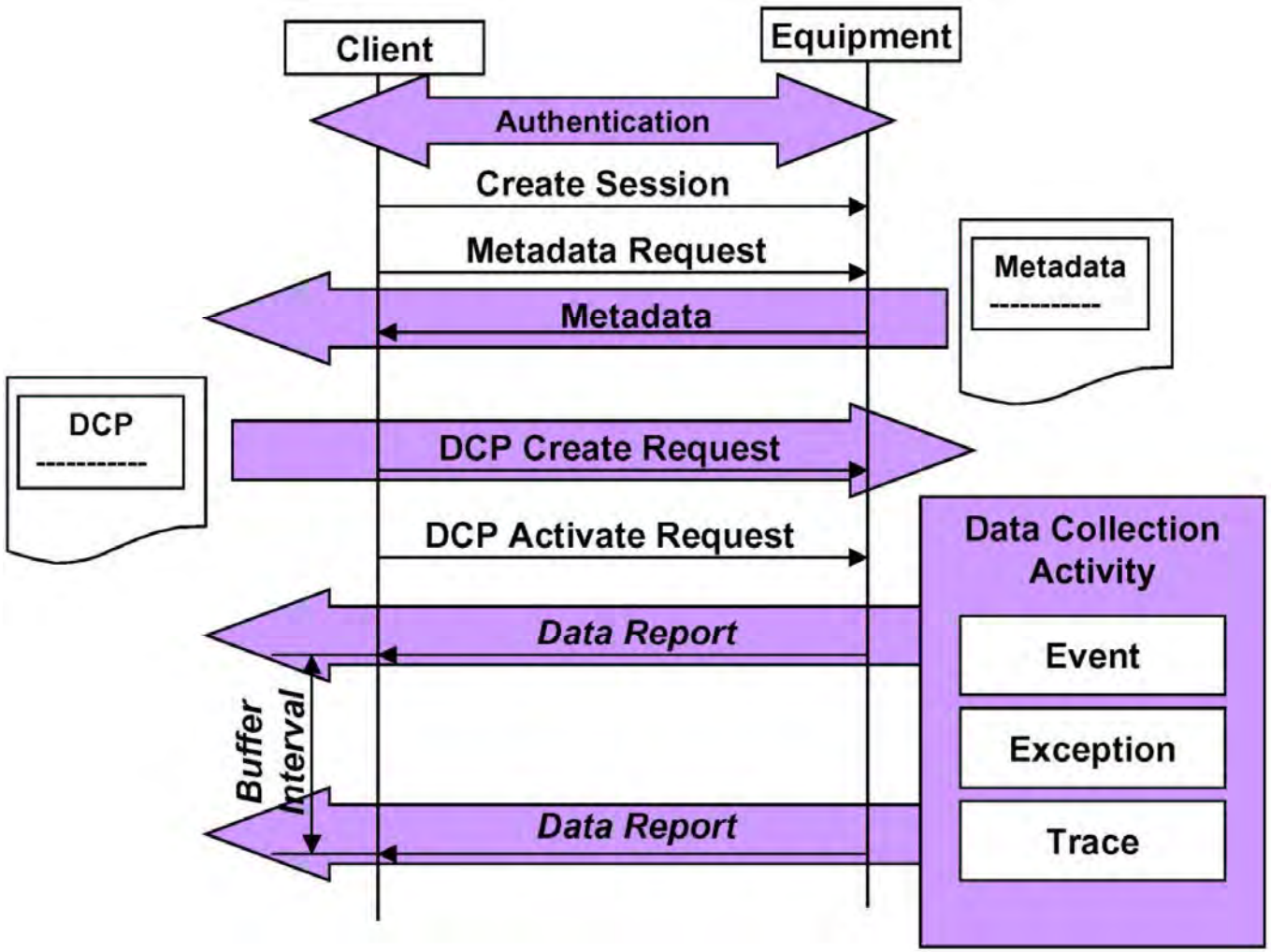
SEMI EDA Standards



EDA Standards on Equipment



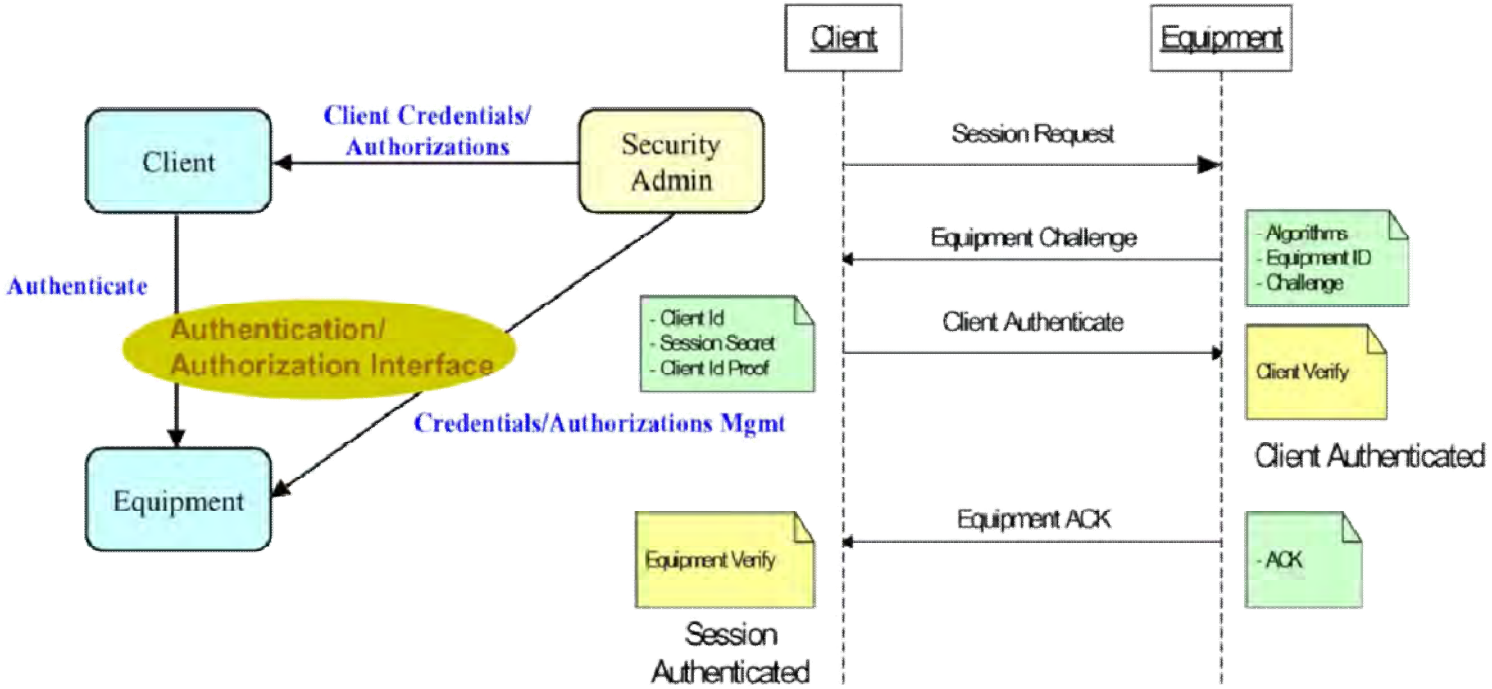
EDA Integrated Scenario



E132 – Authentication & Authorization

- **Purpose**
  - Provide a means for factories to control, in Software, which application are permitted to communicate with the equipment
  - Provide a means for factories to control, in software, which equipment services application are permitted to use
- **Scope**
  - EDA communication technologies only (SECS-II communication will not be addressed)

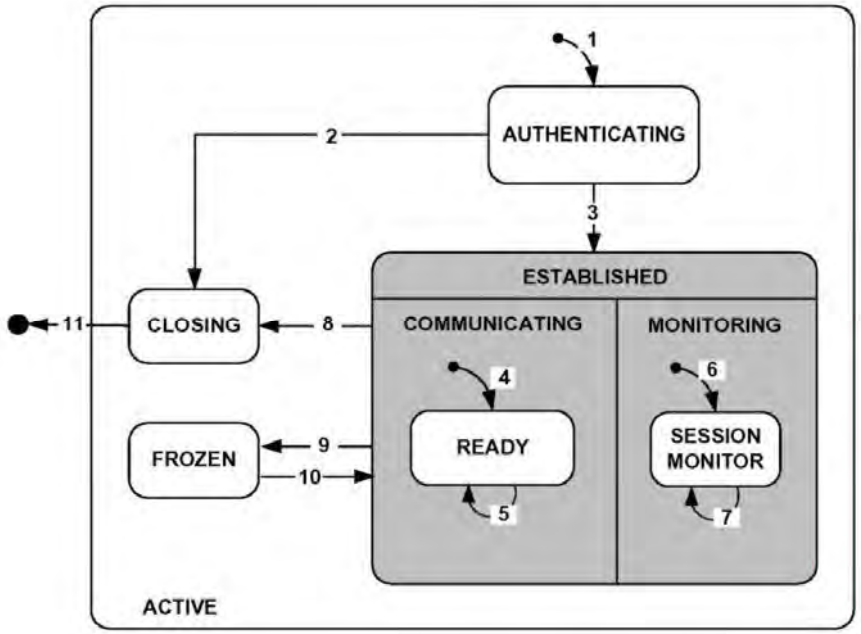
E132 Authentication Model



- **Central Security Admin**
  - **Implemented by factory, assigns credentials to applications**
- **Client**
  - **Establishes session with equipment, provide credentials**
- **Equipment**
  - **Challenges client for credentials, denies session if not accepted**

E132 Session Management

- Equipment communication scoped by “sessions”
  - Session is created if client successfully authenticates
- Client sends messages to begin and end sessions
  - Only the admin has the privilege to end any client’s session
- Sessions can be ‘persisted’ across shutdowns
  - Facilitates notification to clients of equipment availability

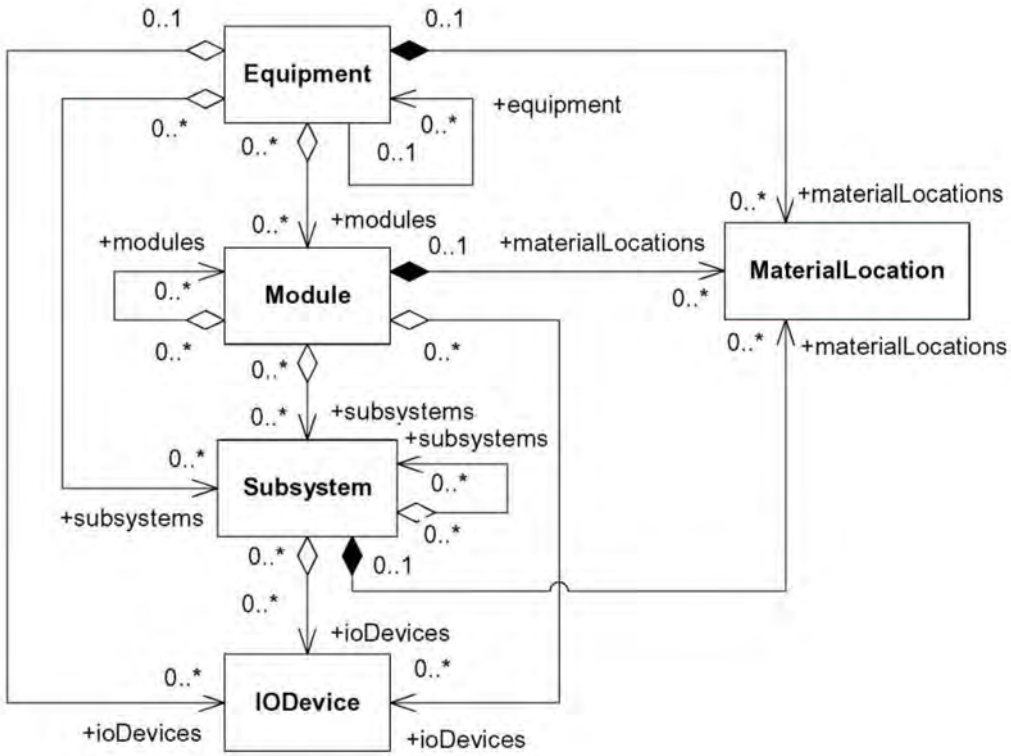


E120 – Common Equipment Model

- **Purpose**
  - Provide a means for suppliers to describe the physical structure of their equipment using common attributes and terminology
  - Provide a means for SEMI standards that depend on information about equipment structure to have a basis for doing so using common attributes and terminology
- **Scope**
  - Includes constructs for modeling linked equipment, multi-chamber equipment, etc. down to the actuator/sensor level

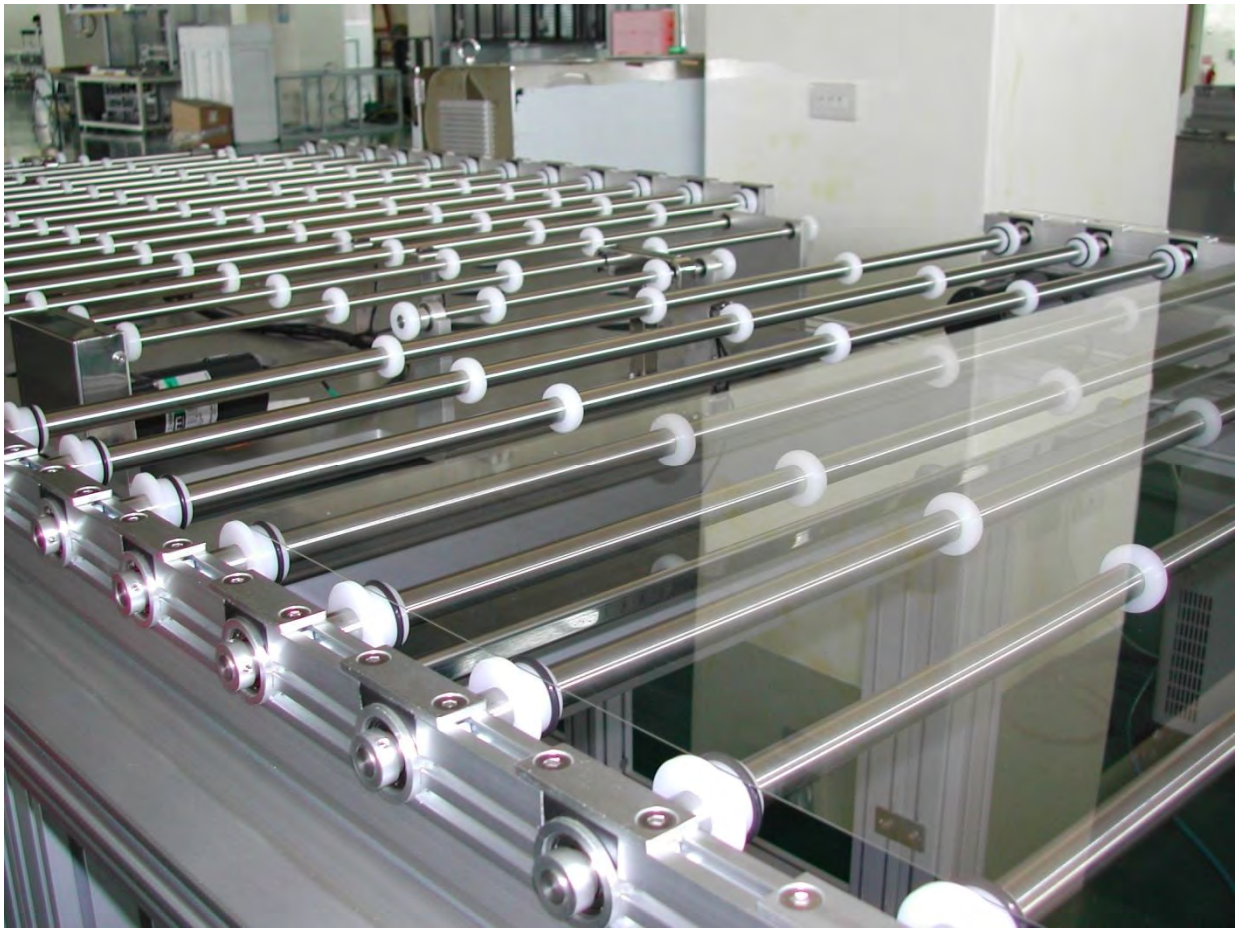
CEM Composition Model

- **Hierarchical model**
- **Allows 'nesting' (e.g., modules within modules)**
- **Supports description of low-level sensors, higher-level processing modules, and the overall equipment**



CEM Example

- **Conveyor**

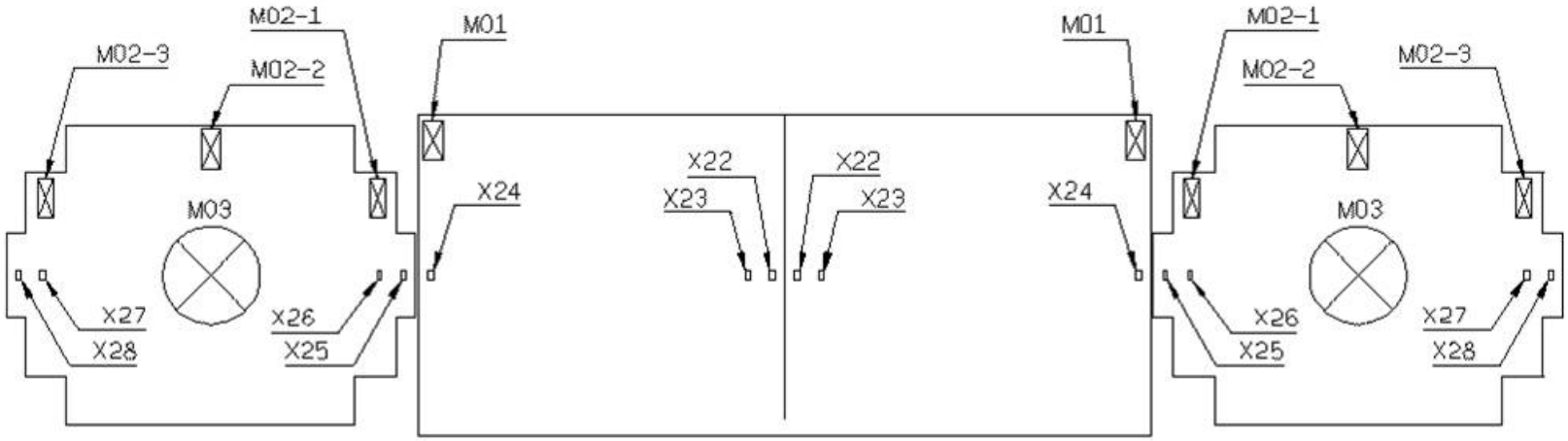


CEM Example

- **Conveyor Sensor & Motor**

CEM Example

- **Conveyor**

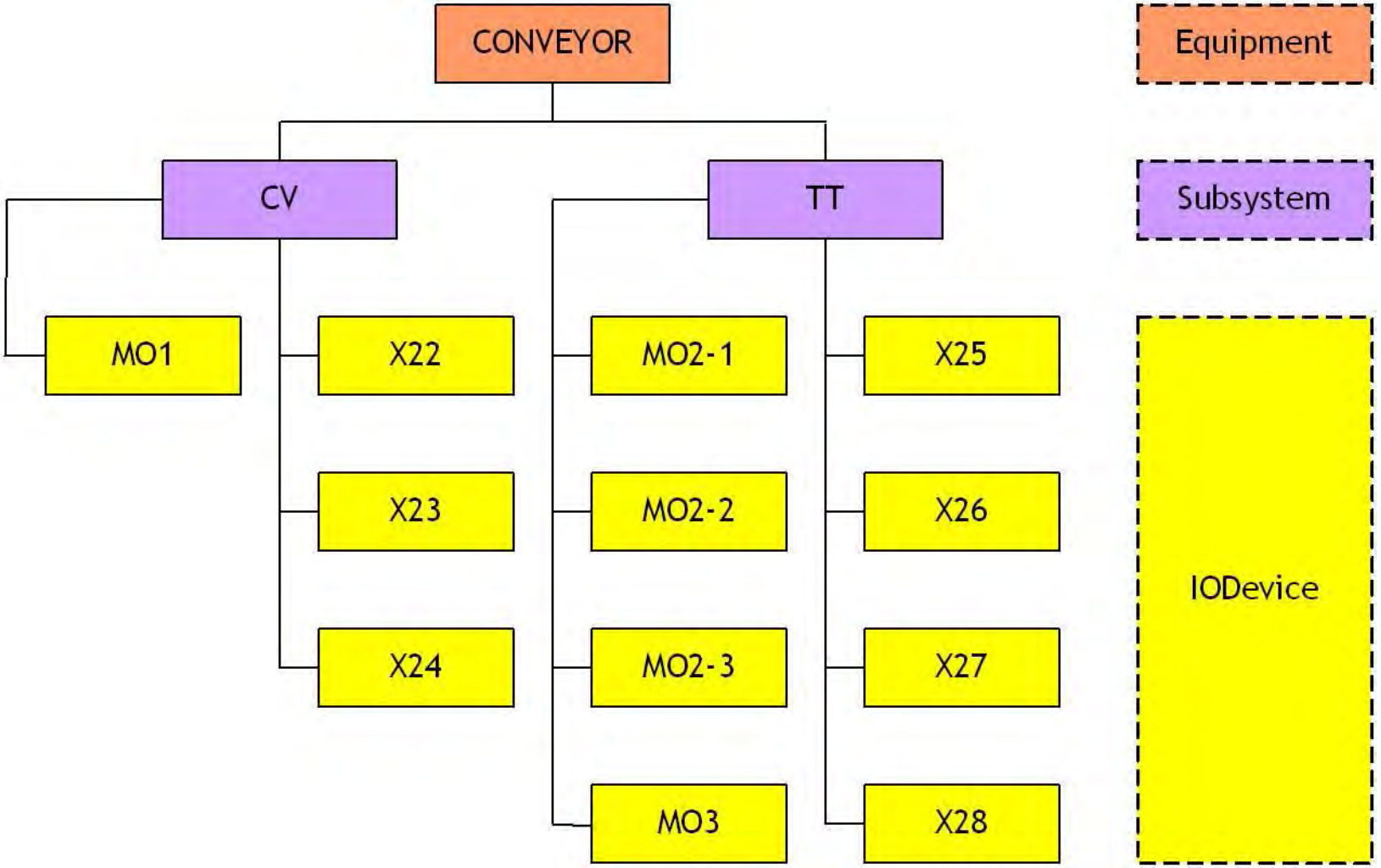


GLASS : 1100X1300X0.5T
GLASS PASSLINE : 1400MM
CONVEYOR :
DRIVE - INDUCTION MOTOR + INVERTER
SPEED - 10-20MM/SEC
TURN TABLE :
DRIVE - DD MOTOR
SPEED - 10SEC/180°
CONVEYOR SENSOR : SLOW/STOP SENSOR

TT-A : 旋轉台-A
CV-A : 輸送帶-A
TT-B : 旋轉台-B
CV-B : 輸送帶-B

M01:固定式輸送帶傳動馬達
M02:旋轉台輸送帶傳動馬達
M03:旋轉台旋轉馬達
X22:固定式輸送帶停止SENSOR
X23:固定式輸送帶減速SENSOR
X24:固定式輸送帶進入SENSOR
X25:旋轉式輸送帶停止SENSOR
X26:旋轉式輸送帶減速SENSOR
X27:旋轉式輸送帶減速SENSOR
X28:旋轉式輸送帶停止SENSOR

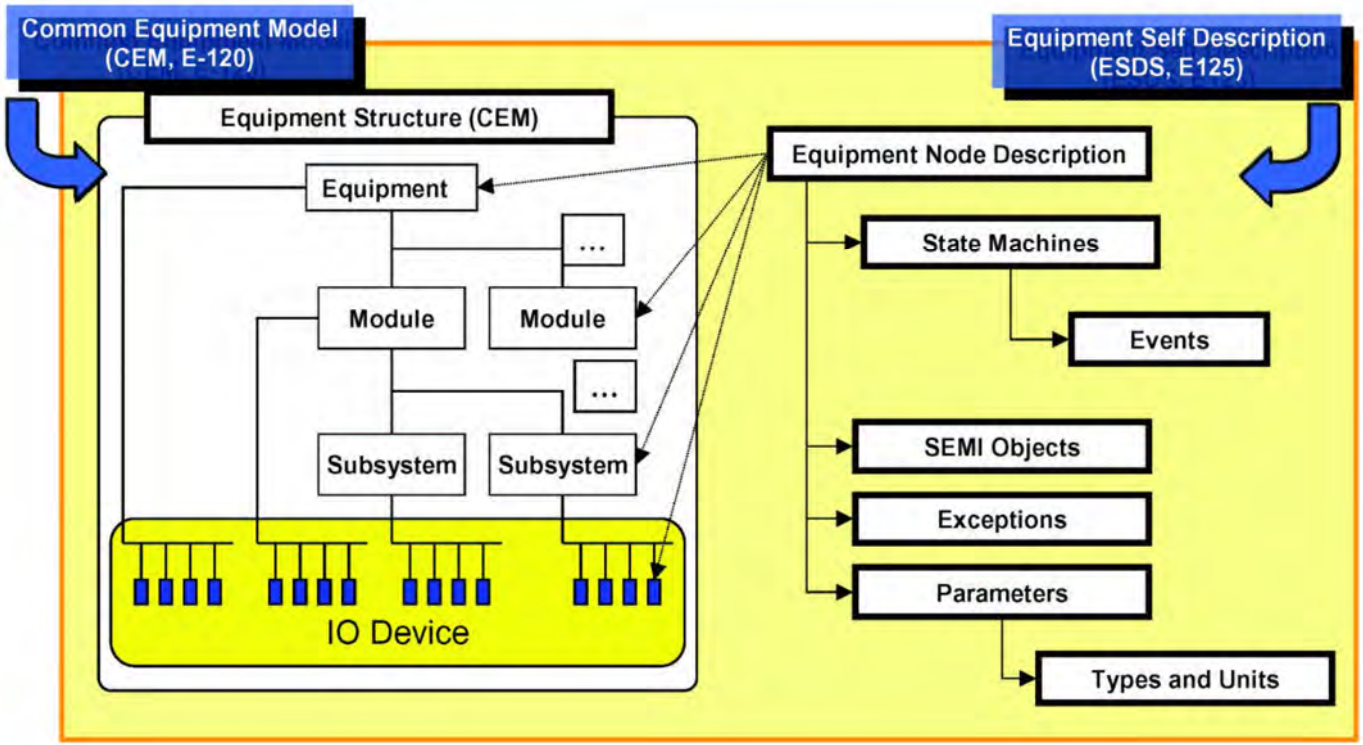
Conveyor CEM Structure



E125 – Equipment Self Description (ESD)

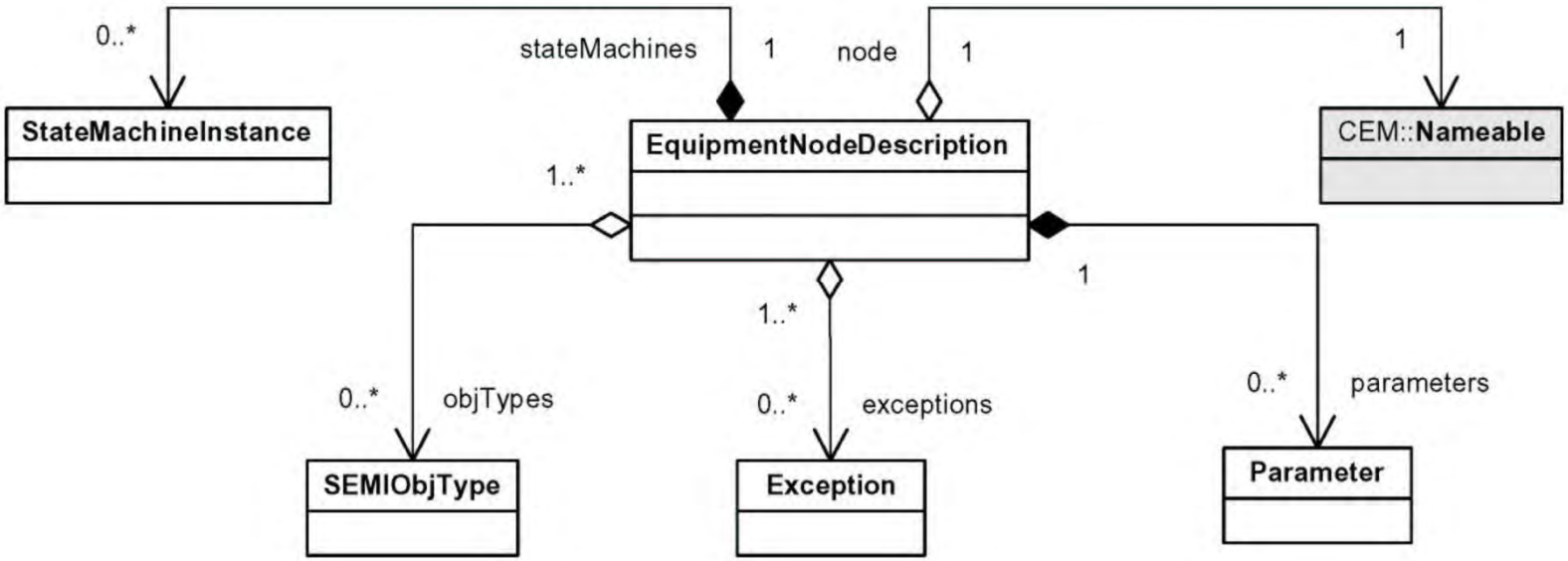
- **Purpose**
  - Provide a means for application to discover via software the physical equipment structure, available data items, events, and exceptions
- **Scope**
  - Includes type description, units, equipment structure (via CEM/E120), supplier-defined and SEMI-defined state models and events, supplier-defined and SEMI-defined alarm and exceptions, SEMI ObjTypes (events and attributes only), data/configuration/control parameters

E125 Equipment Metadata



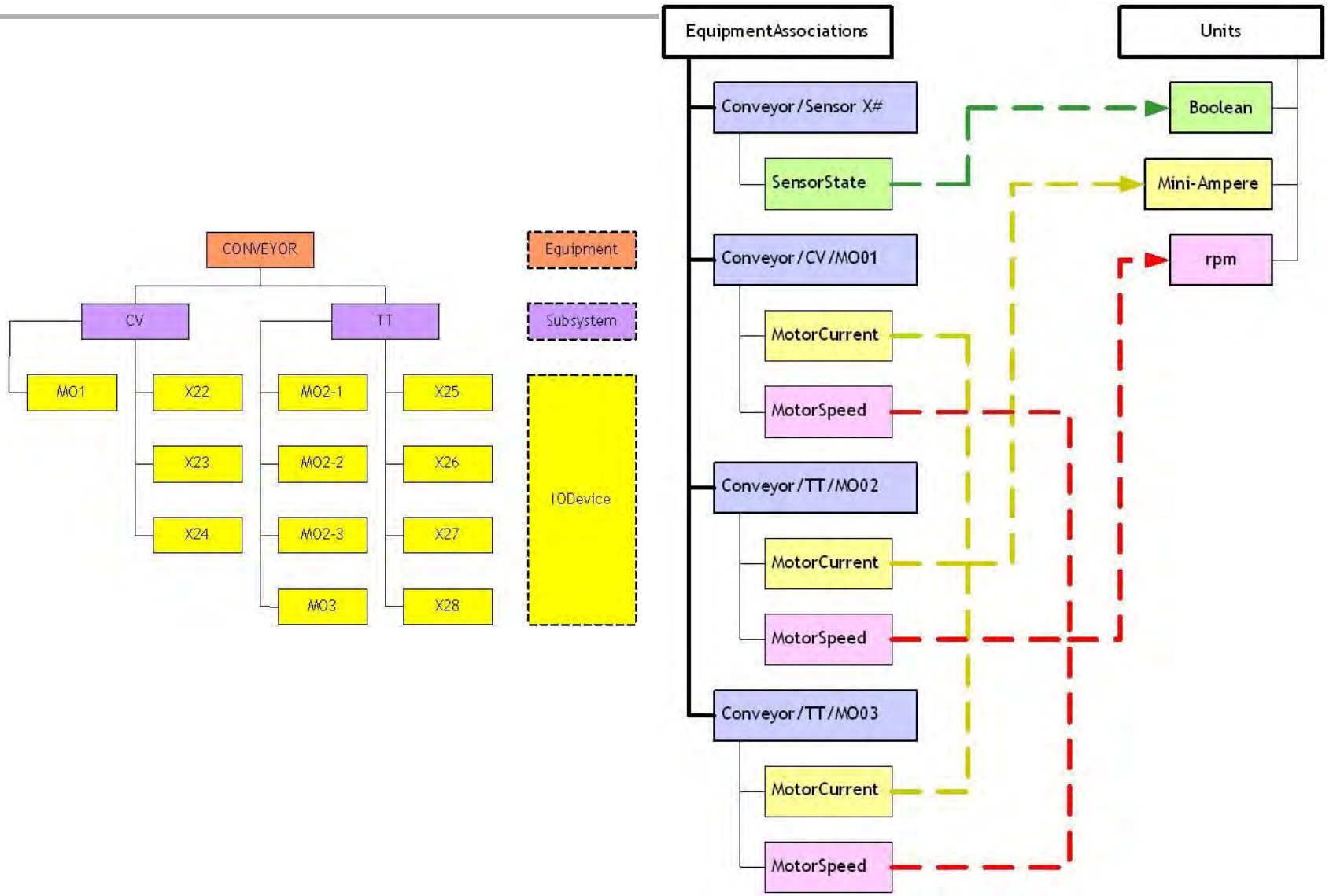
- **Metadata is modularized**
  - Units, types, equipment structure, exceptions, etc. are defined independently of one another
- **Associations with equipment structure are centralized**
  - Each equipment node that can produce data is referenced by name (UUID)
  - The data types, events, exceptions, etc. that an equipment node can produce are associated with that node by reference to a unique id

Equipment Node Description



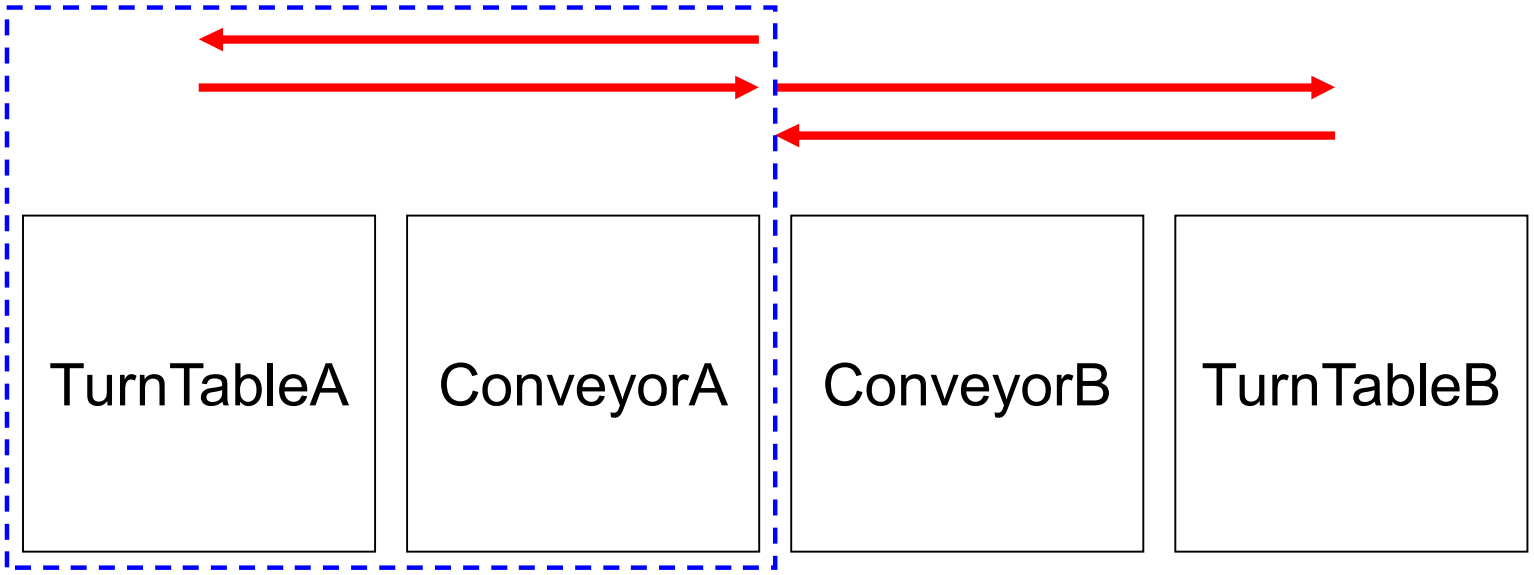
- **CEM::Nameable identifies the CEM node being described**
- **Parameter identifies the parameters available from the CEM node**
- **Exception identifies the exceptions available from the CEM node**
- **SEMIObjType identifies the SEMIObjTypes available from the CEM node**
- **StateMachineInstance identifies the StateMachines available from the CEM node**

Parameter Metadata

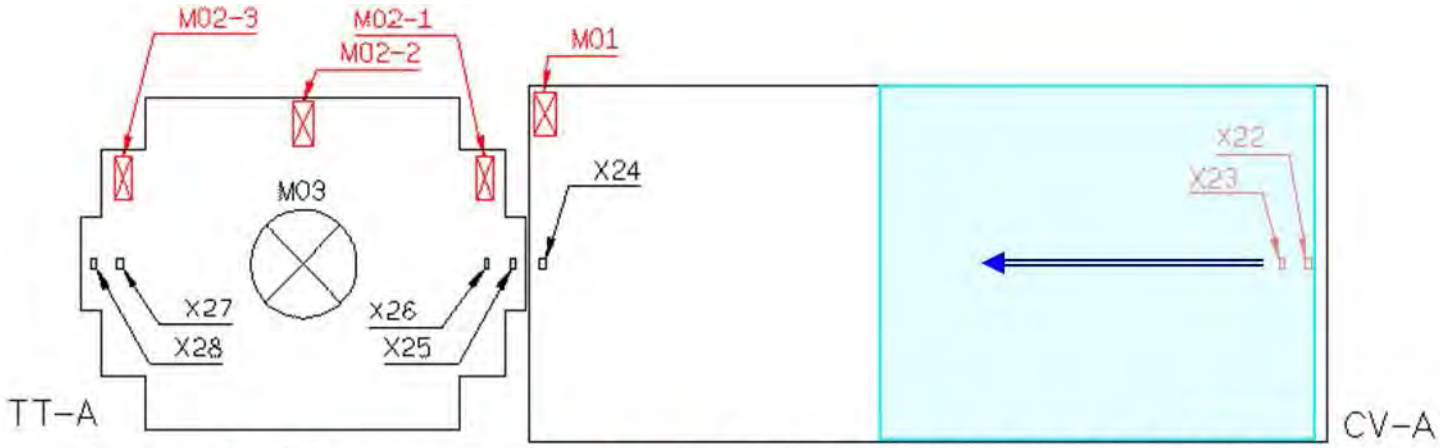


ESD Example : Conveyor

- **Glass transfer sequence on a Conveyor**
  - **CV-A → TT-A → TT-A(TURN) → CV-A → CV-B → TT-B → TT-B(TURN) → CV-B → CV-A…**

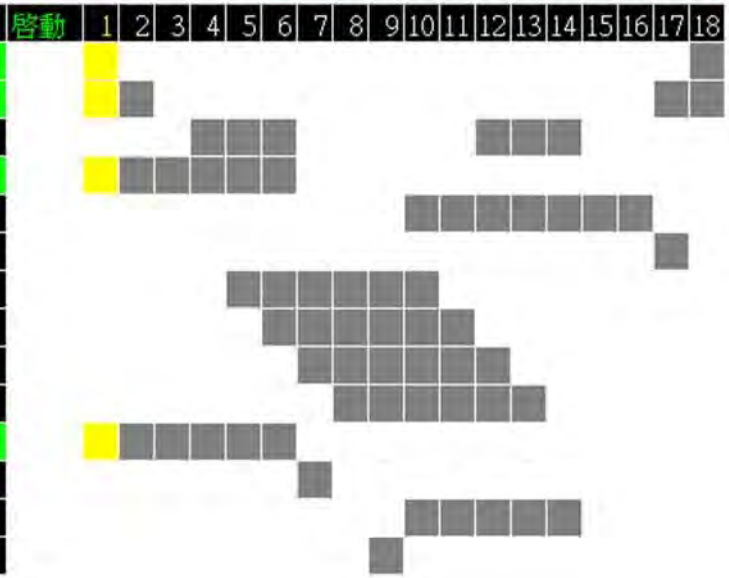


Time Sequence of Conveyor - 1

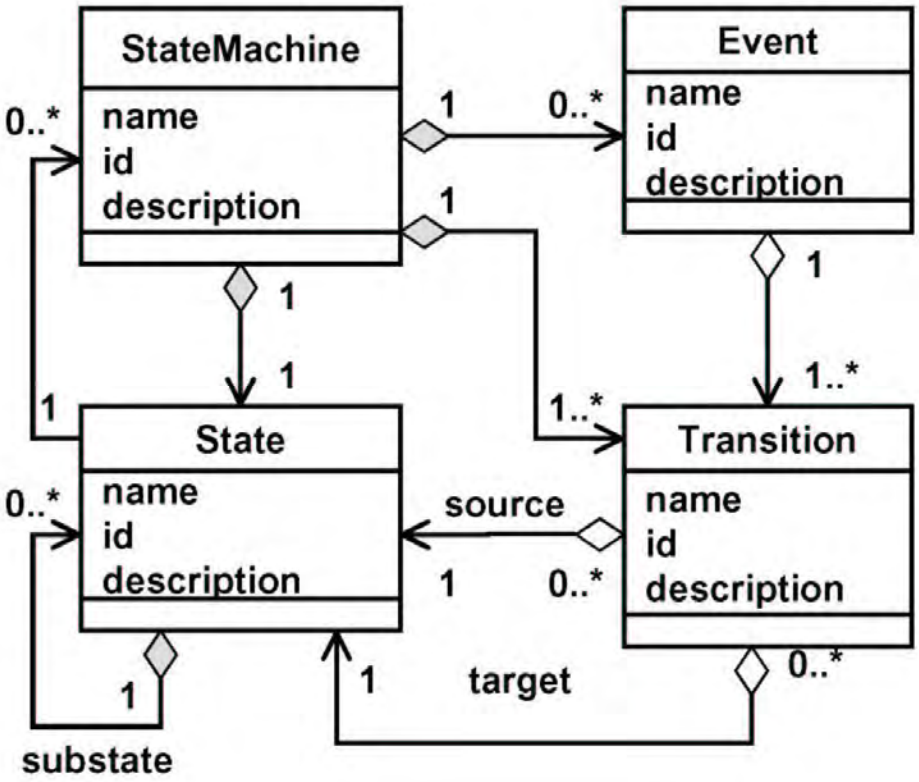


TT-A ： 旋轉台-A
CV-A ： 輸送帶-A

項次	元件編號及說明
1	X22
2	X23
3	X24
4	M01-CW-High/Motor
4	M01-CCW-High/Motor
4	M01-CCW-Low/Motor
5	X25
6	X26
7	X27
8	X28
9	M02_CW-High/MOTOR
10	M02_CW-Low/MOTOR
11	M02_CCW-High/MOTOR
15	M03-CW/MOTOR

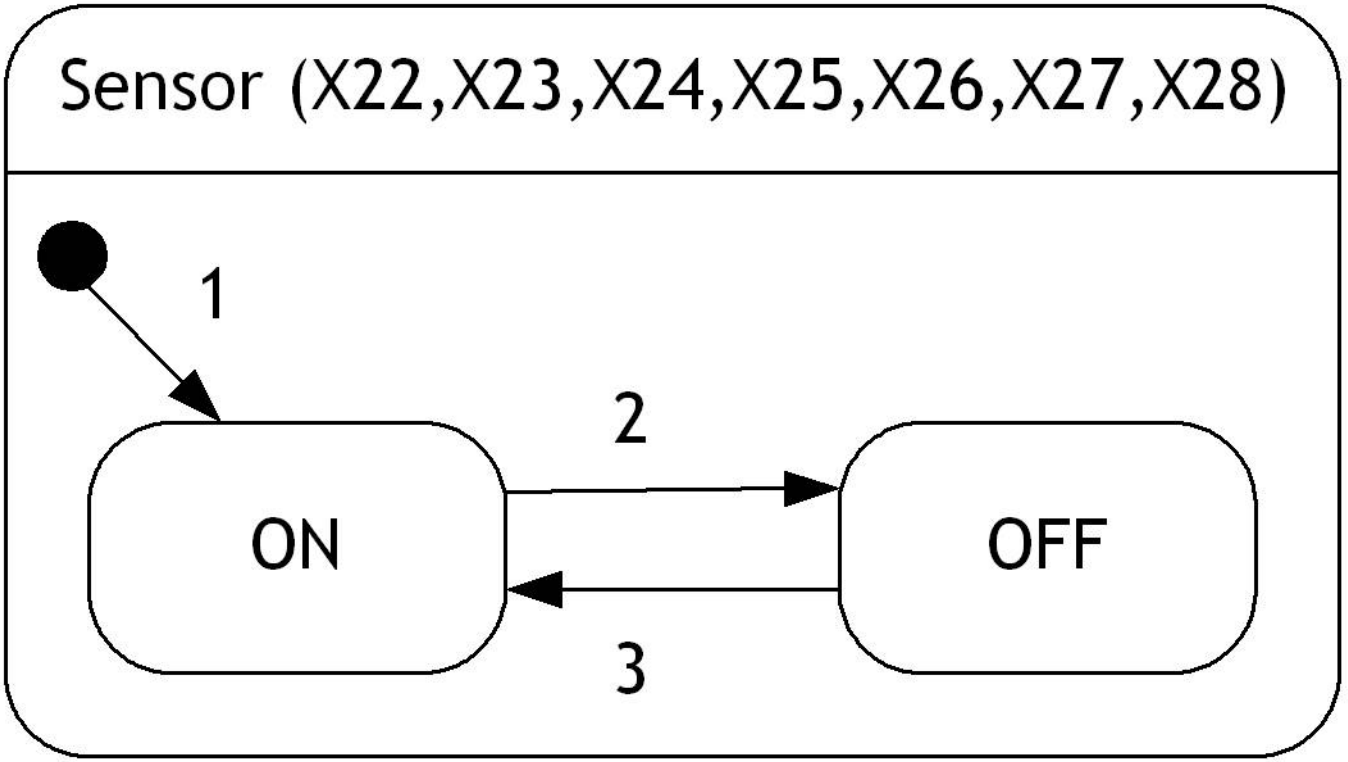


State Machine

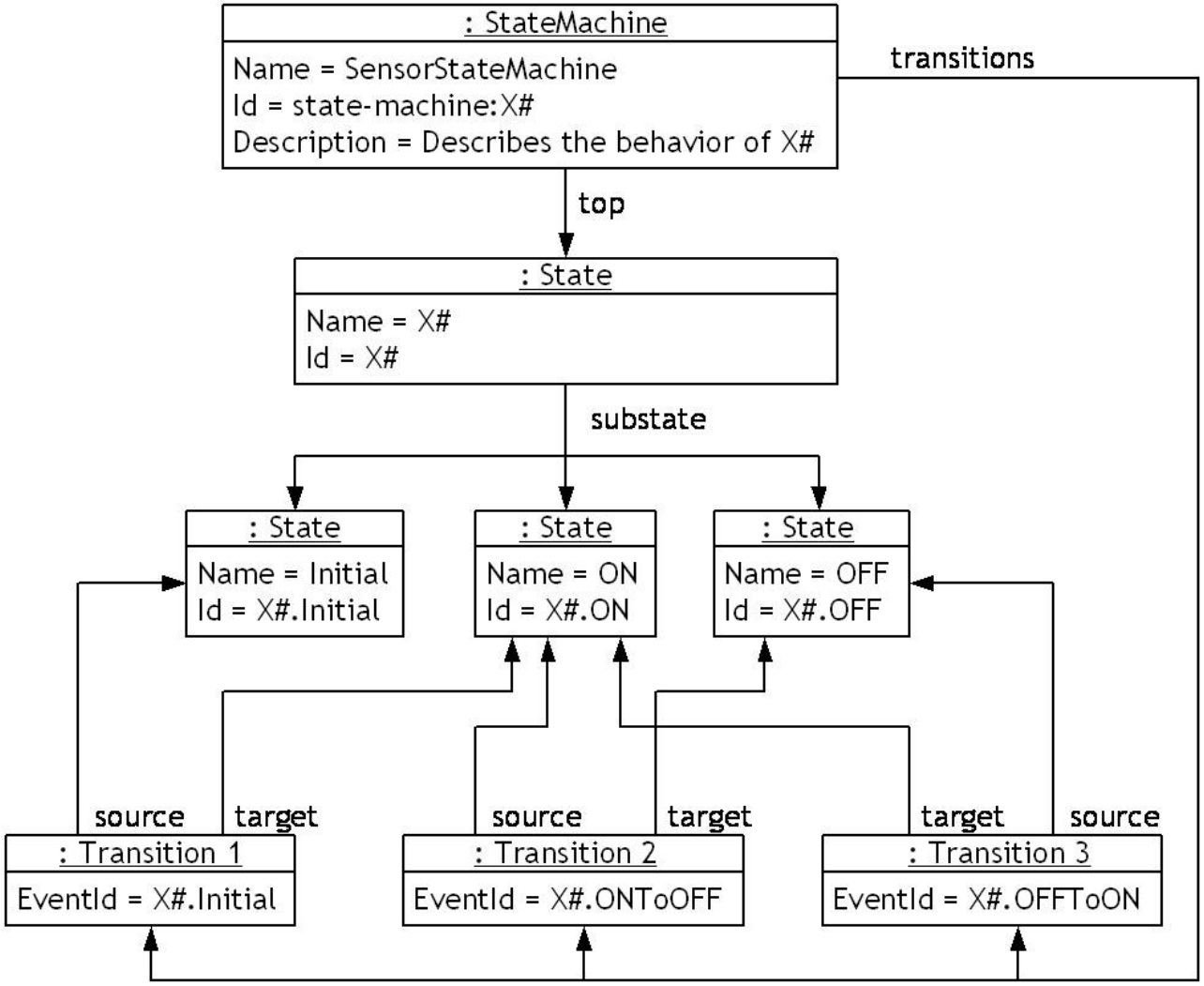


An Event may correspond to one state transition or multiple state transitions.

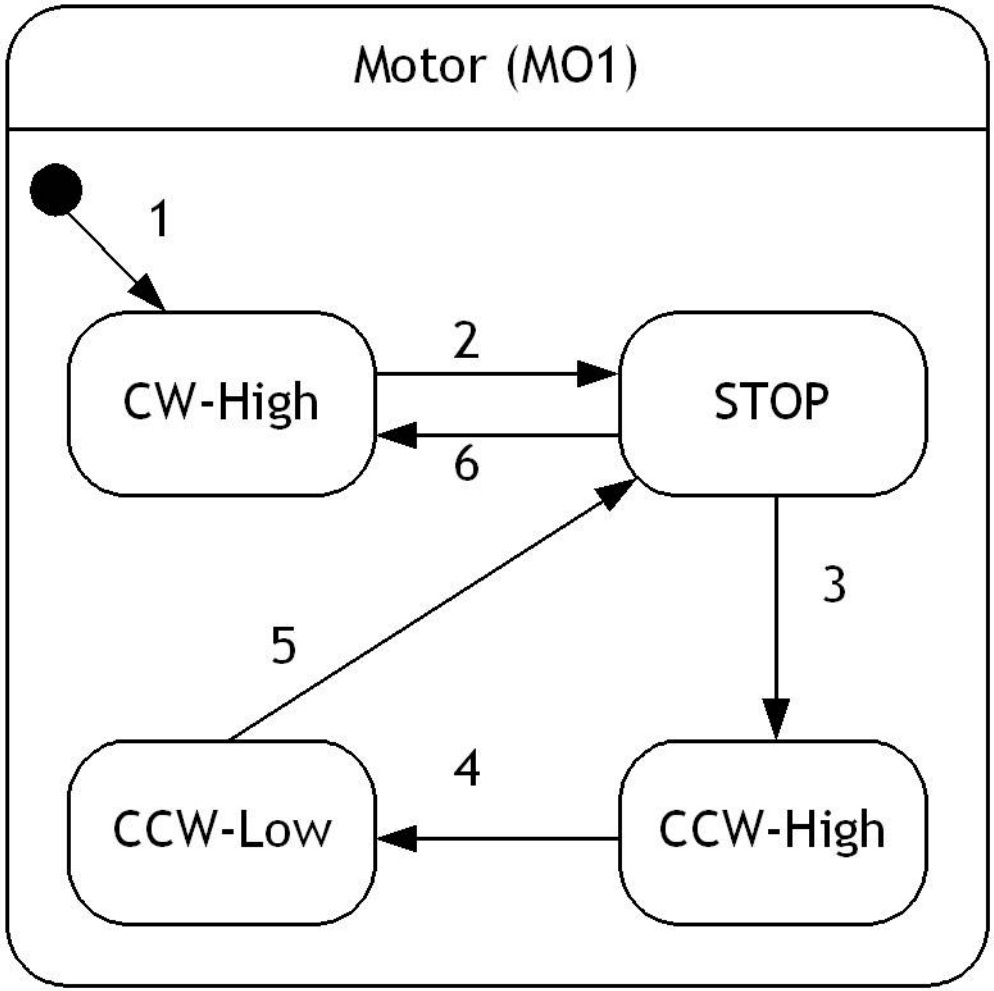
Sensor State Model



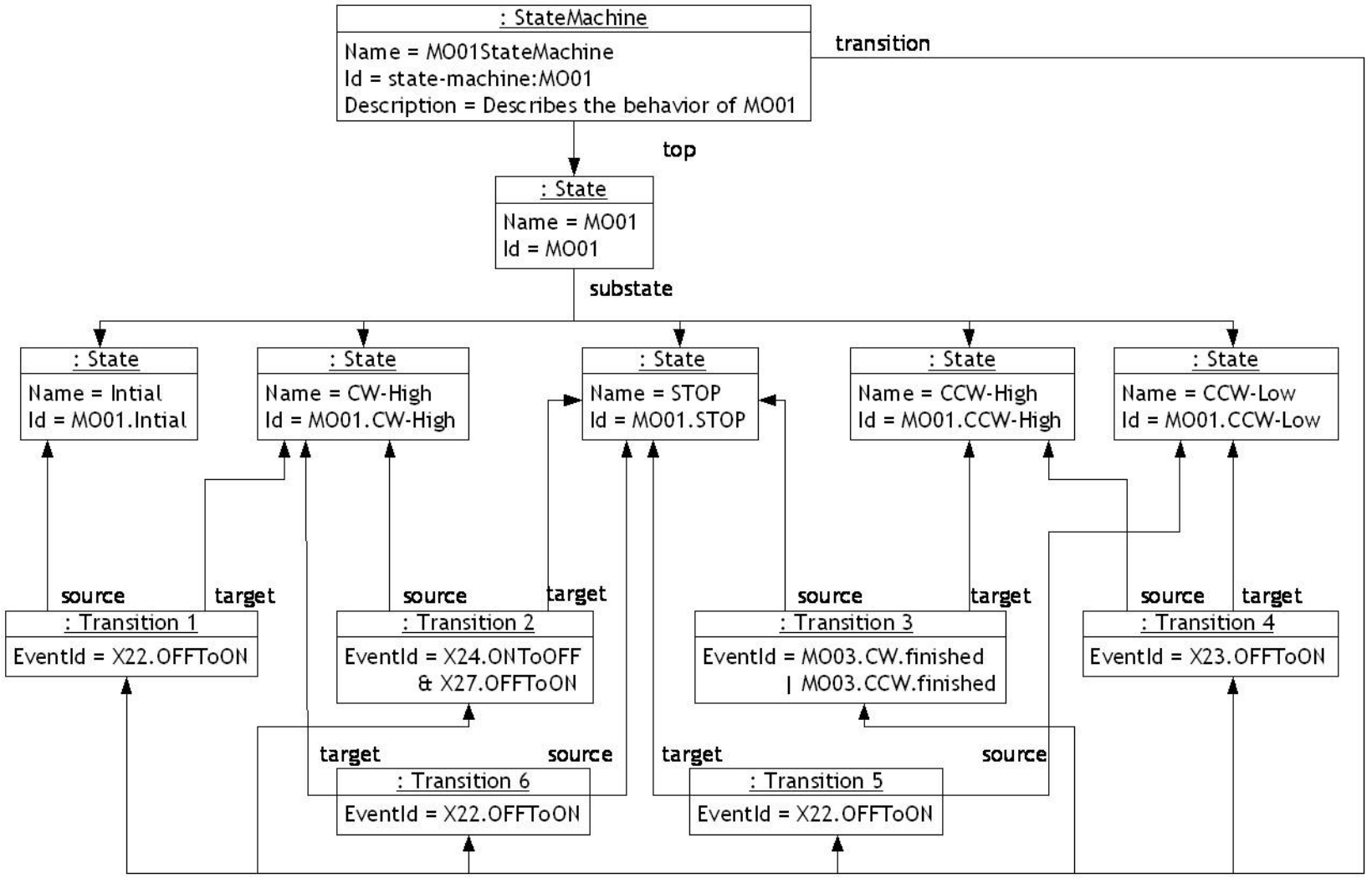
Sensor State Machine Metadata



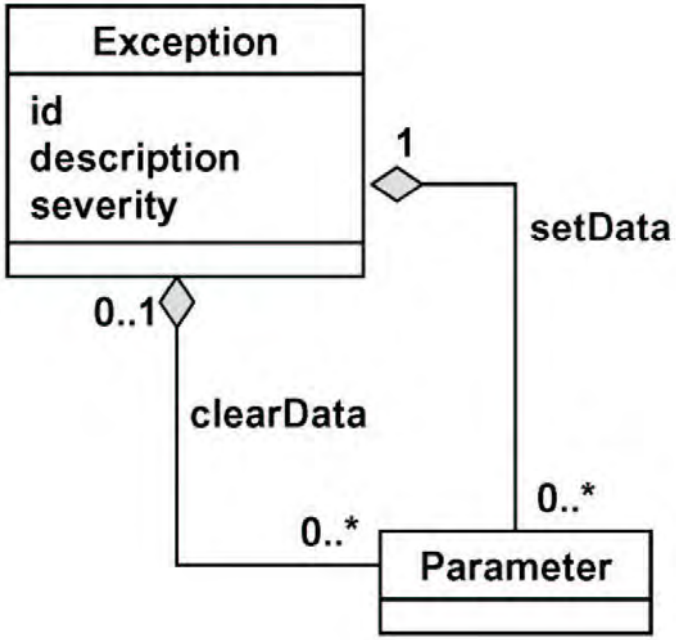
Motor (MO1) State Model



Motor (MO1) State Machine Metadata



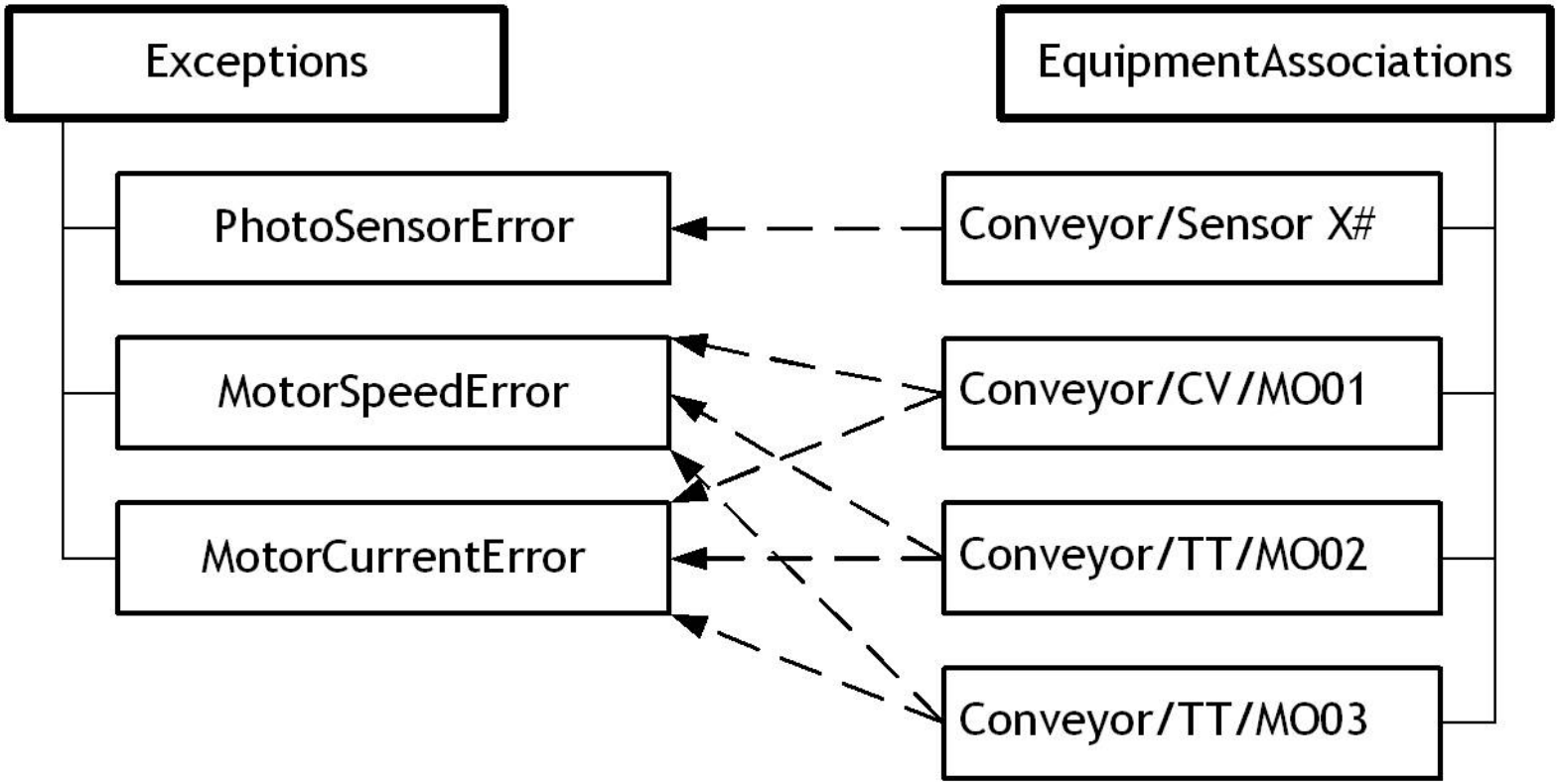
Exception



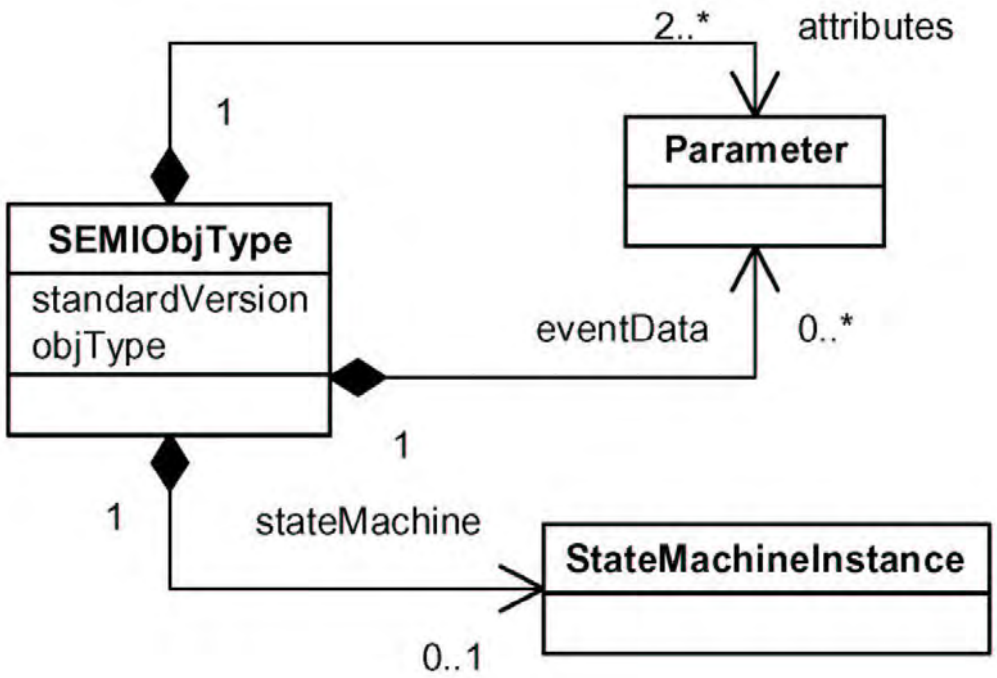
"setData" defines parameters that can be reported when the Exception is set.

"clearData" defines parameters that can be reported when the Exception is cleared.

Exception Metadata

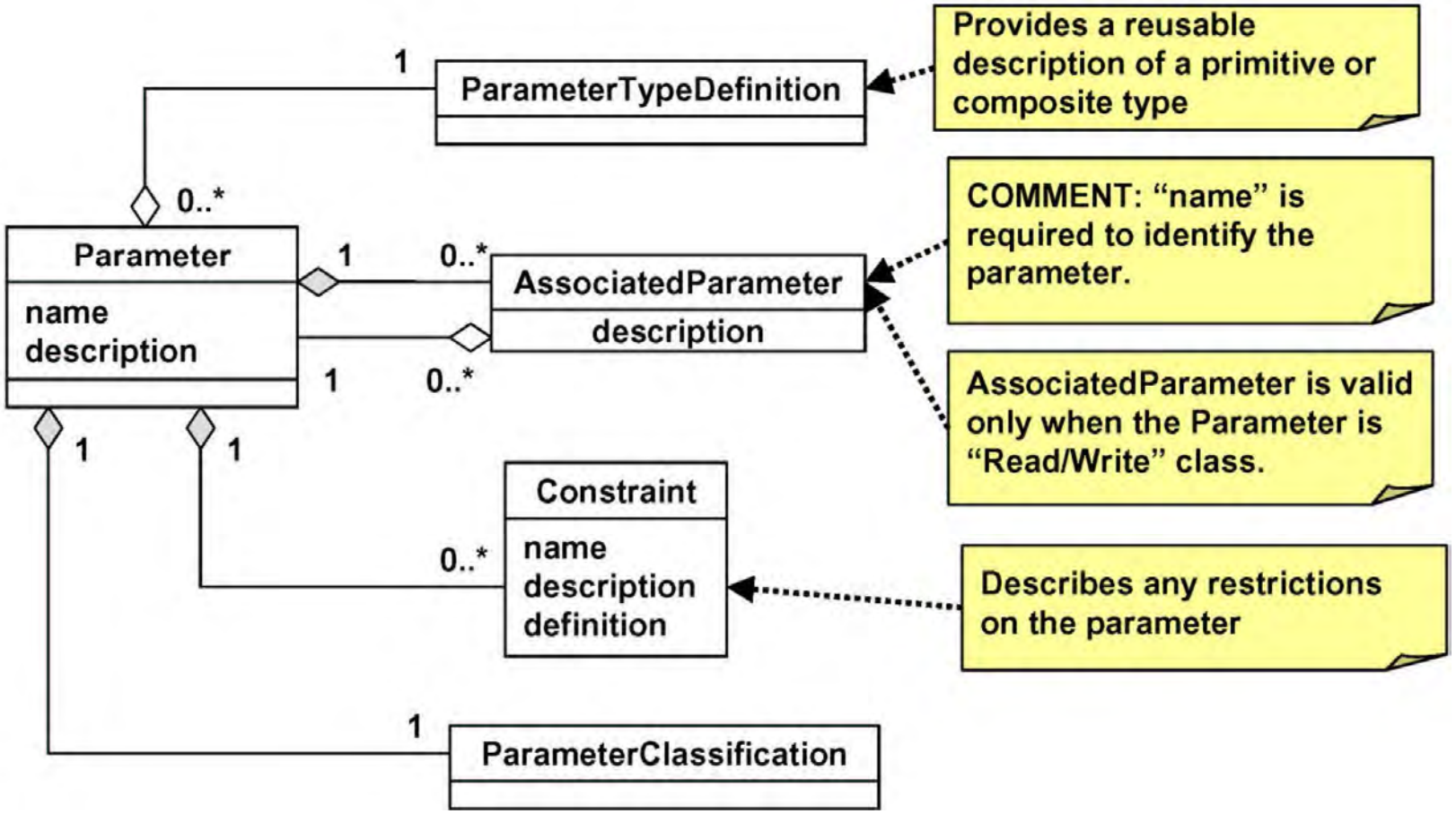


SEMIObjType

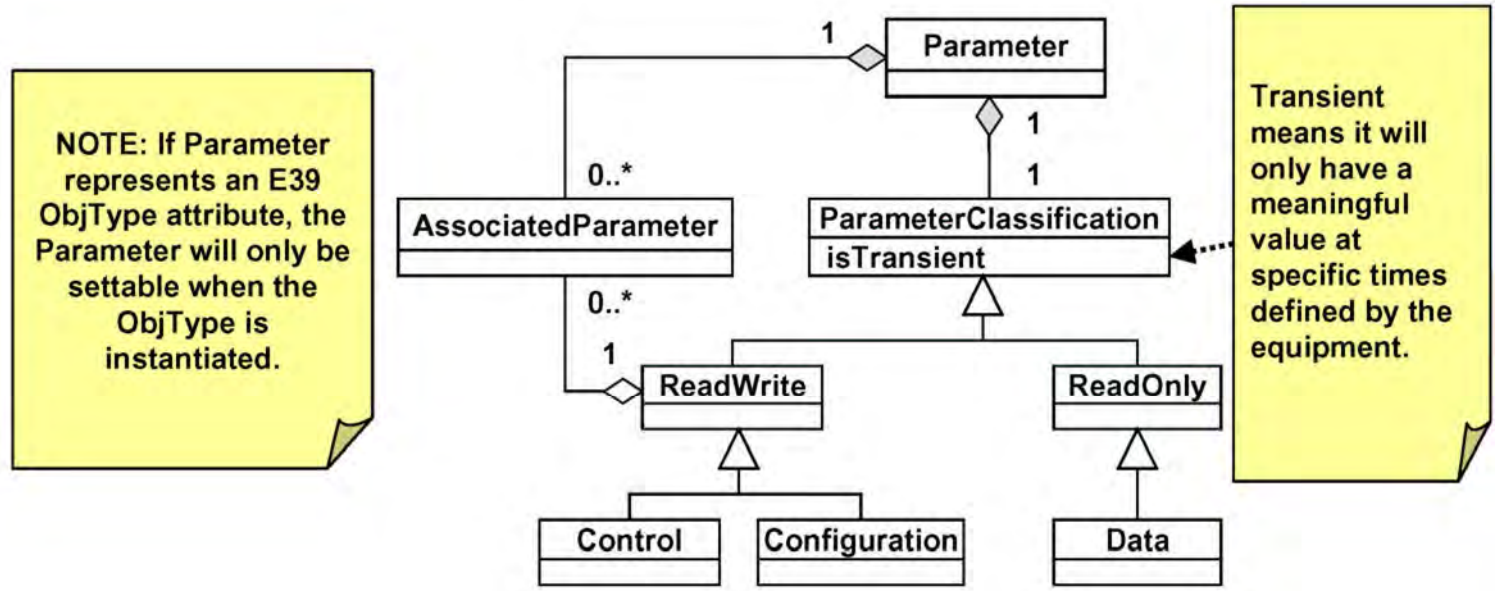


- **Represents a E39 compliant ObjType supported by the equipment**
- **StateMachineInstance is used to define which StateMachine the ObjType has and the data it can generate**
- **Defines the attributes and event data that the object can communicate**

Parameter

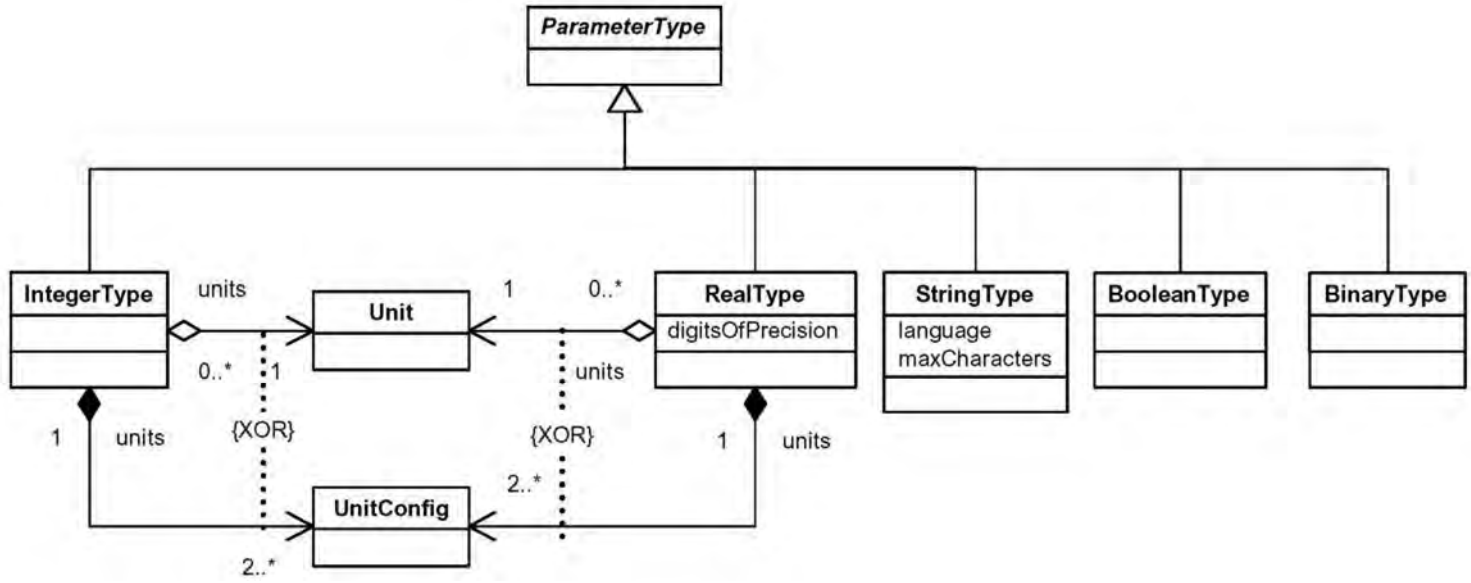

1
ParameterTypeDefinition
Provides a reusable description of a primitive or composite type
0..*
Parameter
name
description
1
0..*
AssociatedParameter
description
COMMENT: "name" is required to identify the parameter.
1
0..*
AssociatedParameter is valid only when the Parameter is "Read/Write" class.
1
1
Constraint
name
description
definition
0..*
Describes any restrictions on the parameter
1
ParameterClassification

Parameter Classification



- **A parameter can either be a ReadWrite or a ReadOnly parameter**
  - **ReadOnly parameters represent Data**
  - **ReadWrite can either be a Control parameter or a Configuration parameter**
    - **A control parameter directly affects how the material processes, measures, or tests**
    - **A configuration parameter affects equipment behavior but is not a control parameter**

ParameterTypeDefinition



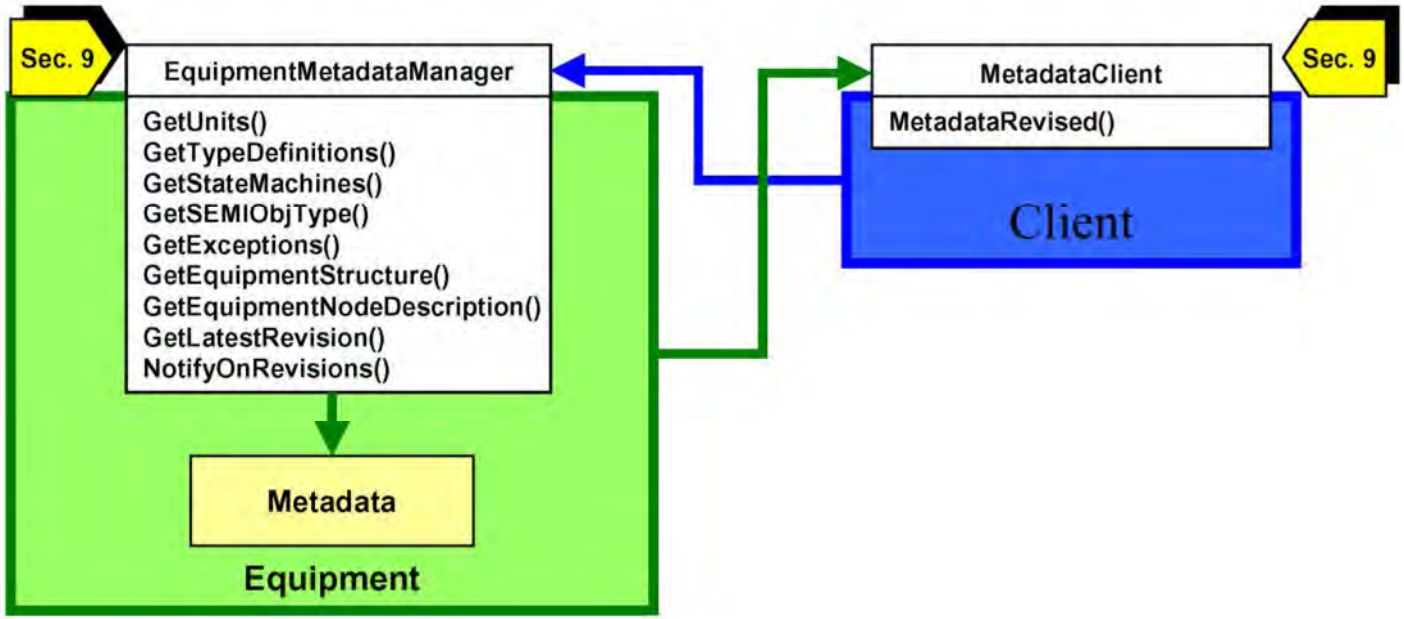
- **The ParameterTypeDefinition permits reuse across parameters of the same datatype.**
- **It also supports structures and arrays**

Units

- **Units are used for numeric parameter types**
- **Can also describe dimensionless units**

Interface for Equipment Metadata

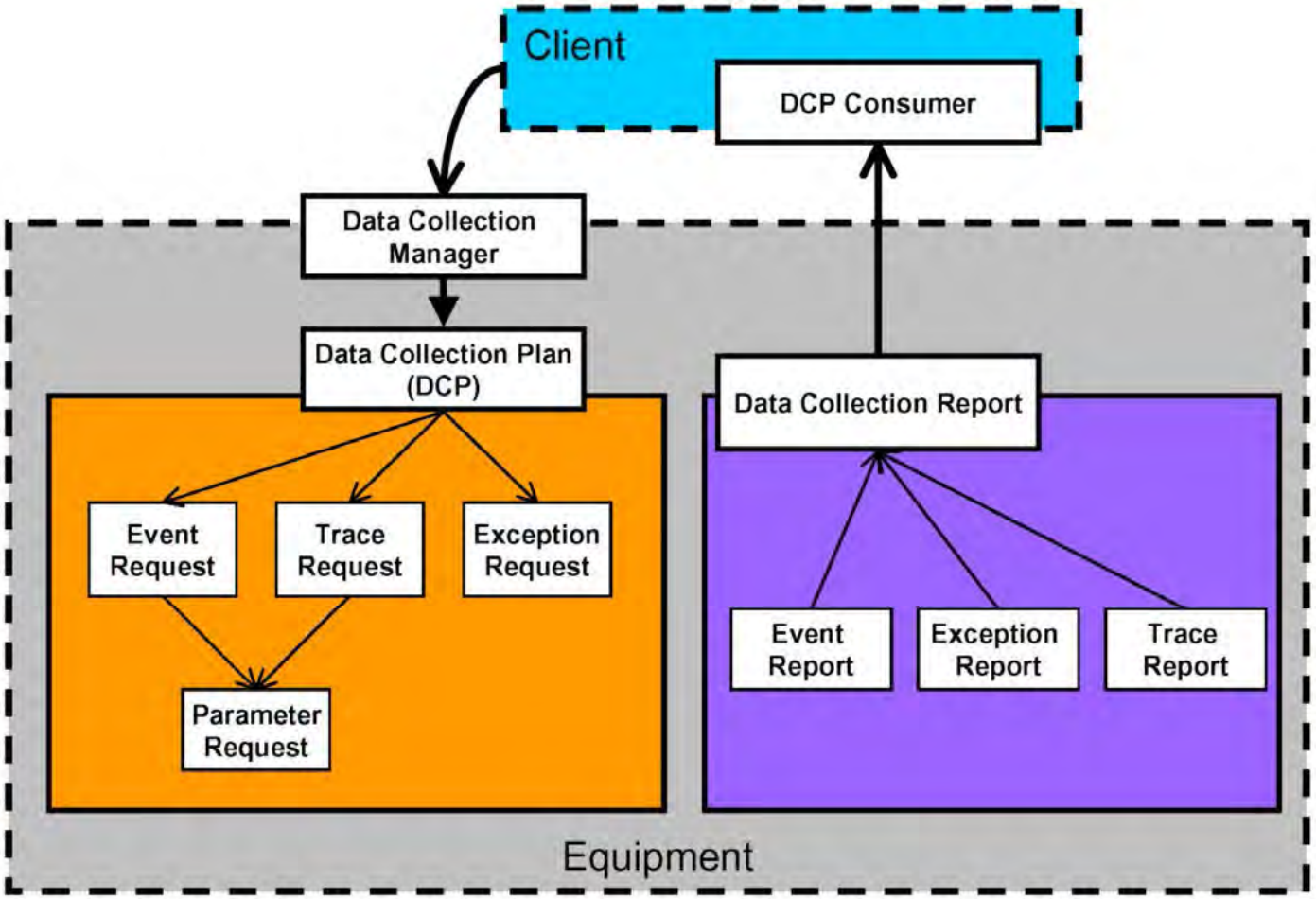
- **Equipment metadata is supplier-sensitive information**
  - **Access to it is via an authenticated session with the equipment**
- **Metadata interface is modular**
  - **Follows the modularity of each type of metadata**



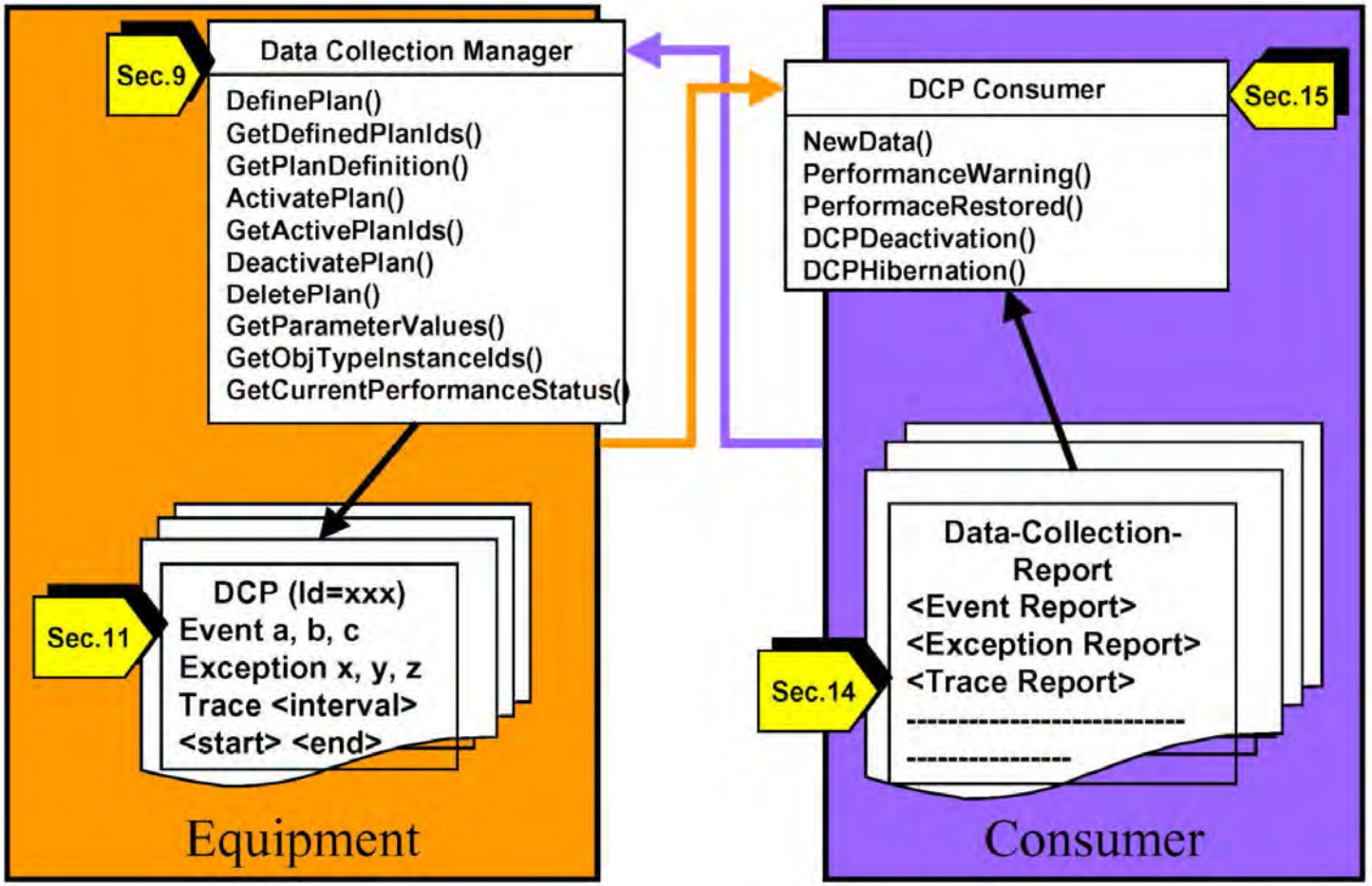
E134 – Data Collection Management

- **Purpose**
  - Provide a means for applications to organize all data needs (trace, exception, event) into logical, named units that can be individually activated and deactivated
- **Scope**
  - Data collection plan definition, DCP management interface, state models, data reporting formats
  - Event-driven “push” style data collection (events, traces, exceptions)
  - On-demand data collection
  - On-tool buffering of collected data
  - Equipment performance warnings
  - DCP management privilege model

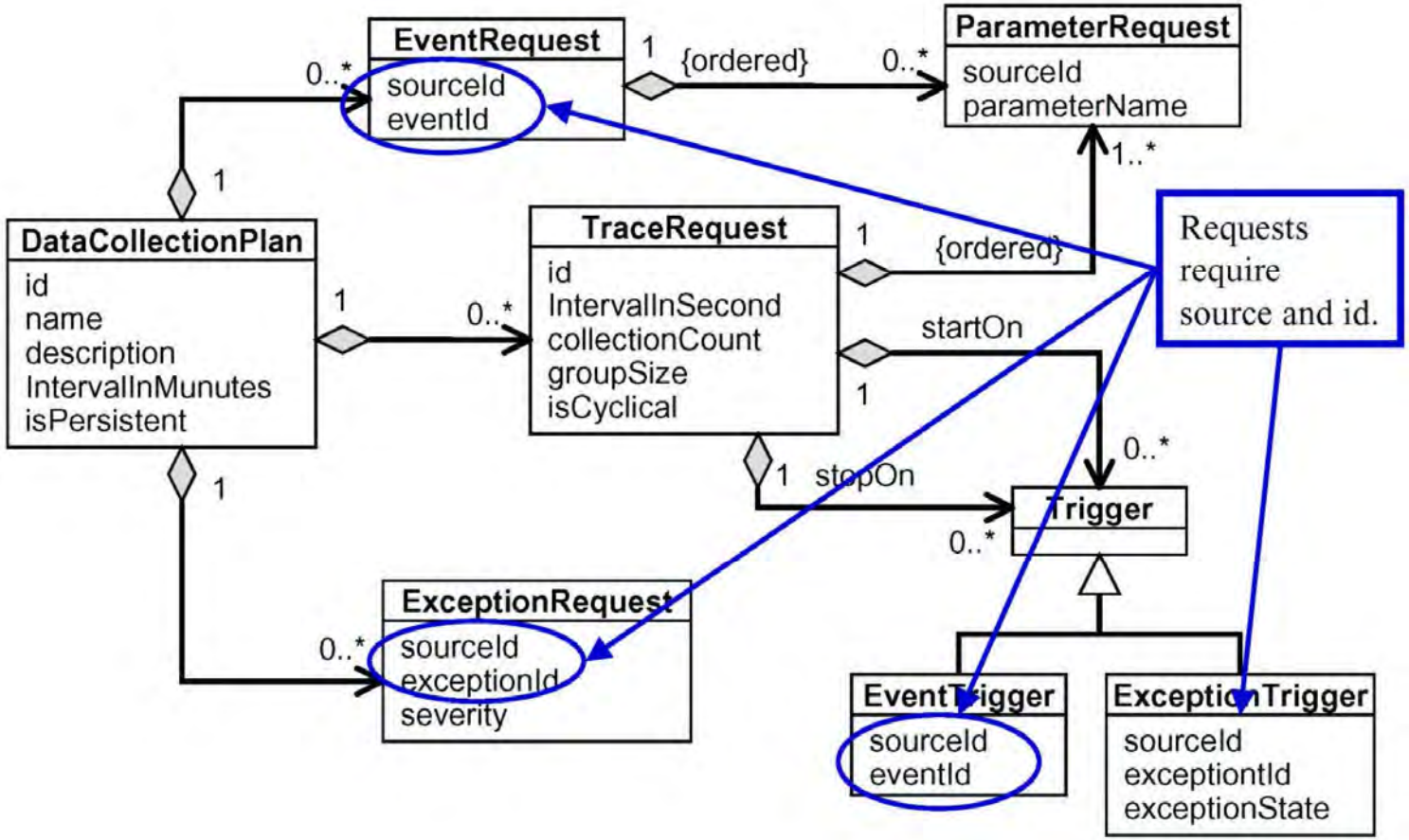
Data Collection - Overview



Manager and Consumer Interface



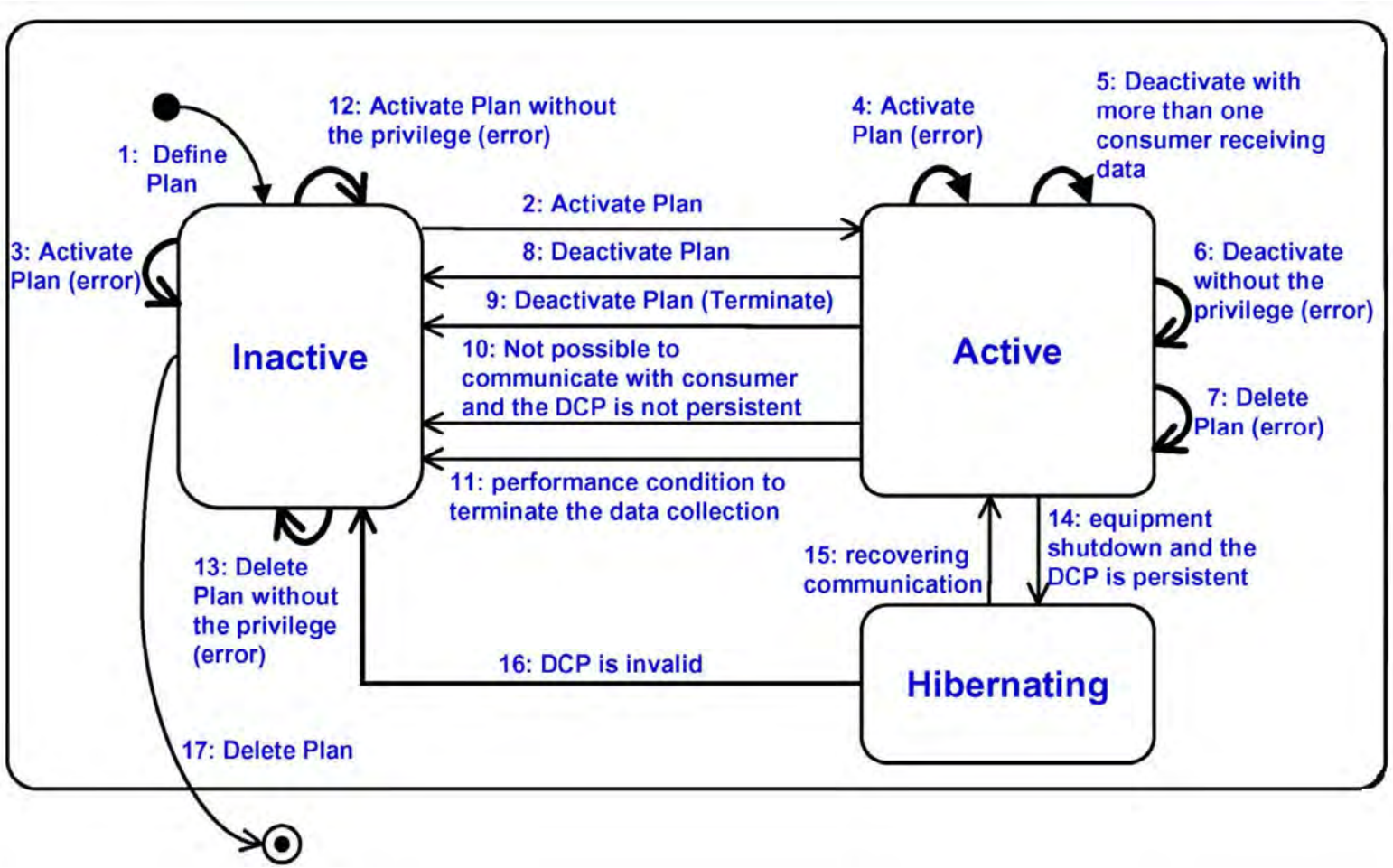
DCP – Data Collection Plan



Built-in DCP

- **Equipment supplier can provide pre-defined DCP's that are included with the equipment, and don't require definition by the consumer.**
- **Built-in DCP's can be activated and viewed by any client using ActivatePlan and GetPlanDefinition**
- **Built-in DCP's cannot be deleted by any consumer.**
- **Reference: *E134 – section 11.1.2.1 Built-in DCP***

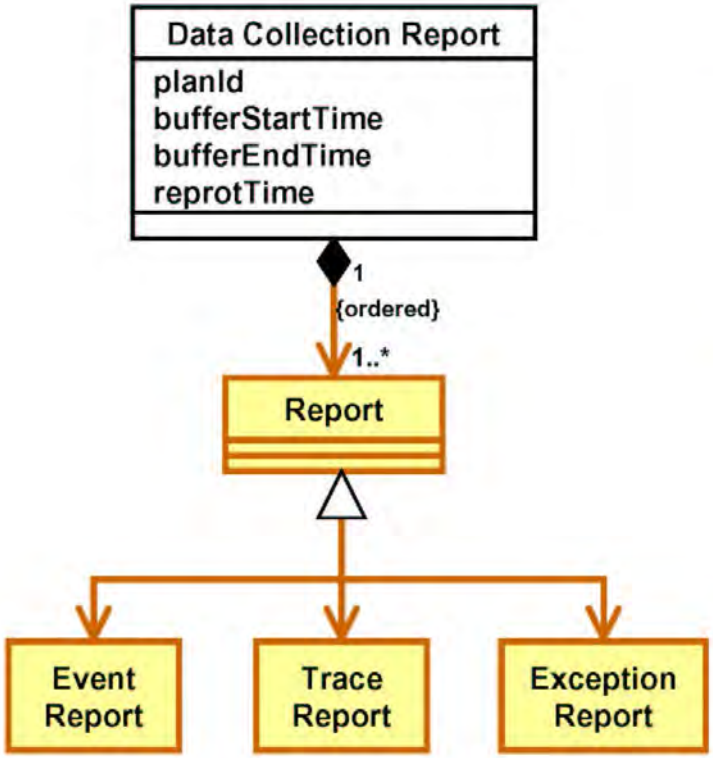
DCP State Model

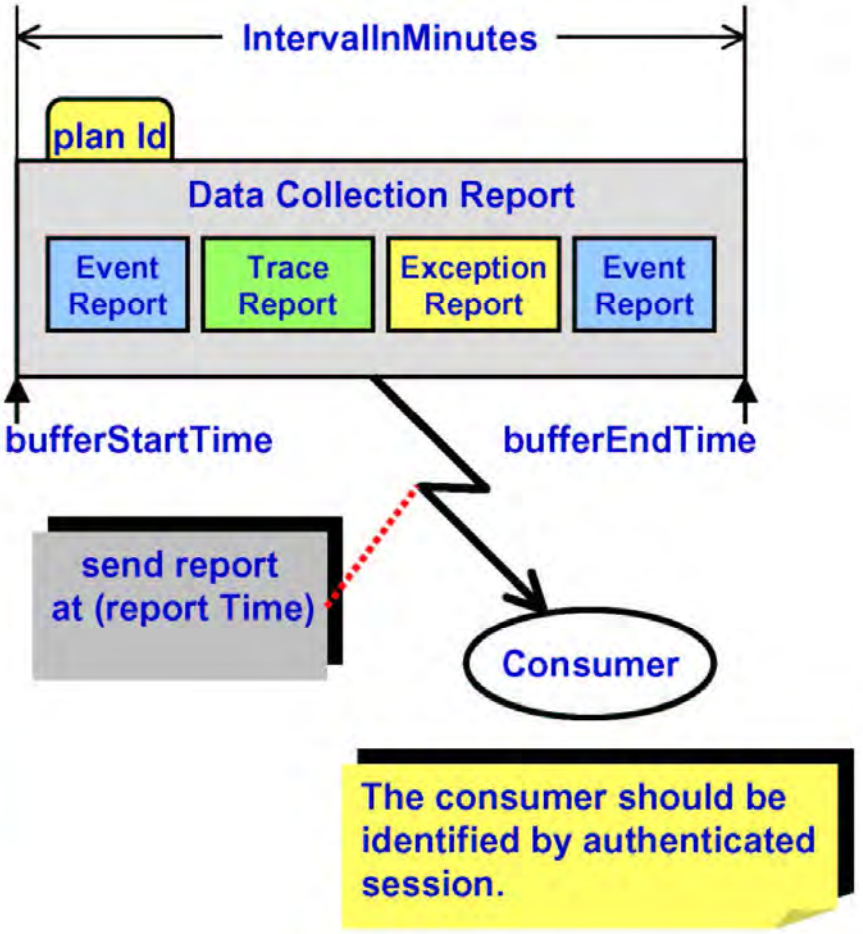


DCP Operation

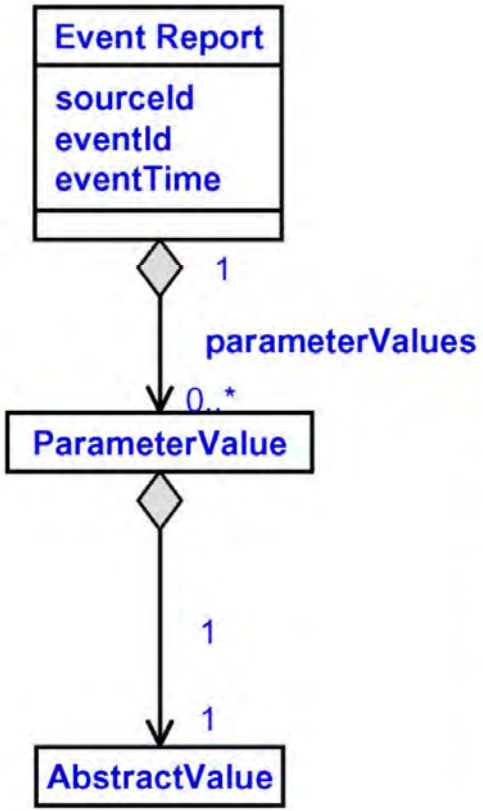
- **DCP Activation/Deactivation/Deletion**
  - Multiple consumers can activate the same DCP simultaneously
  - When a consumer deactivates a DCP, the DCP will remain active for all other consumers currently using the DCP, unless the consumer indicates the DCP should be terminated
    - Consumer must have sufficient privileges to terminate
  - A DCP can't be deleted if the DCP is still active for one or more consumers
- **Editing DCP**
  - DCP cannot be edited
  - When change of DCP required, consumer should delete Defined DCP and define a new DCP

Example of Data Collection Report



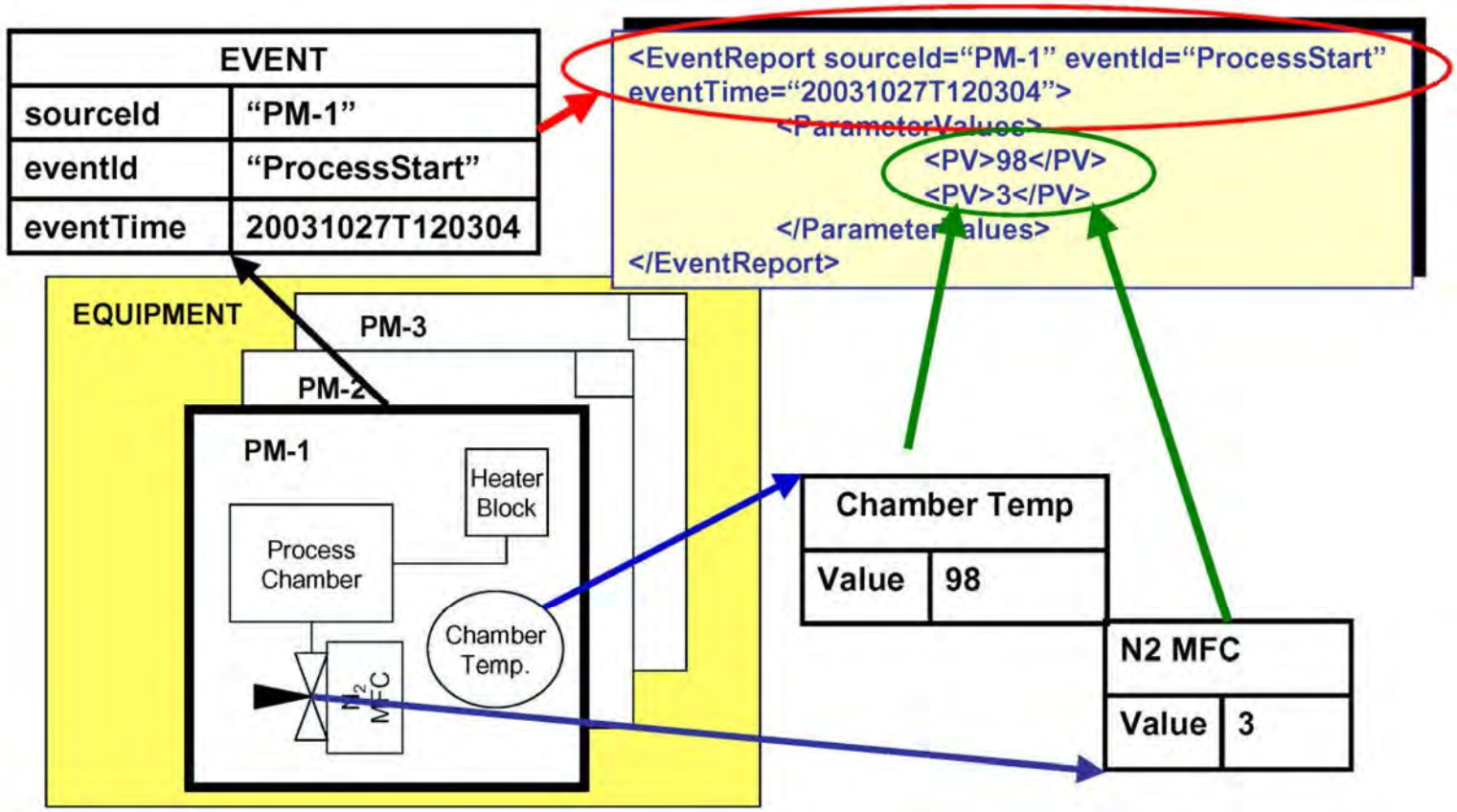


Event Report Definition

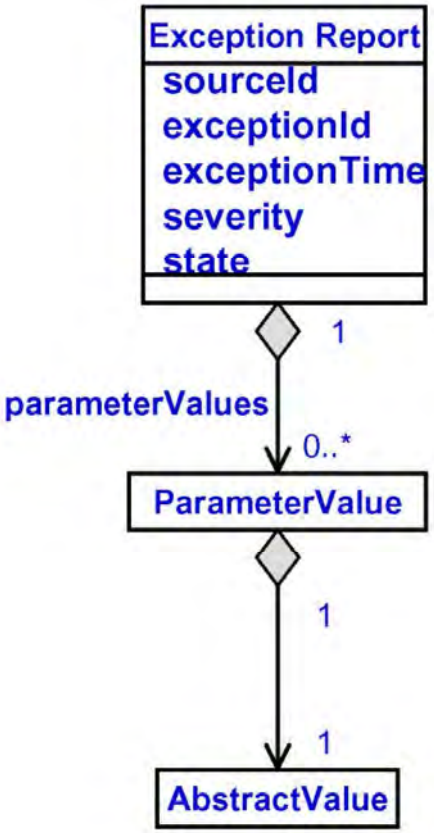


- A Event Report is generated when an event occurs and the consumer requested the Event Report
- "sourceId", "eventId" are used to identify the event
- Event Report may have zero or more ParameterValues
- The ParameterValues appear in the same order as they were listed in the DCP
- ParameterValue provides the actual data value(s)
- AbstractValue represents that here may be a value in a data type; the formats are different in data types

Example for Event Report

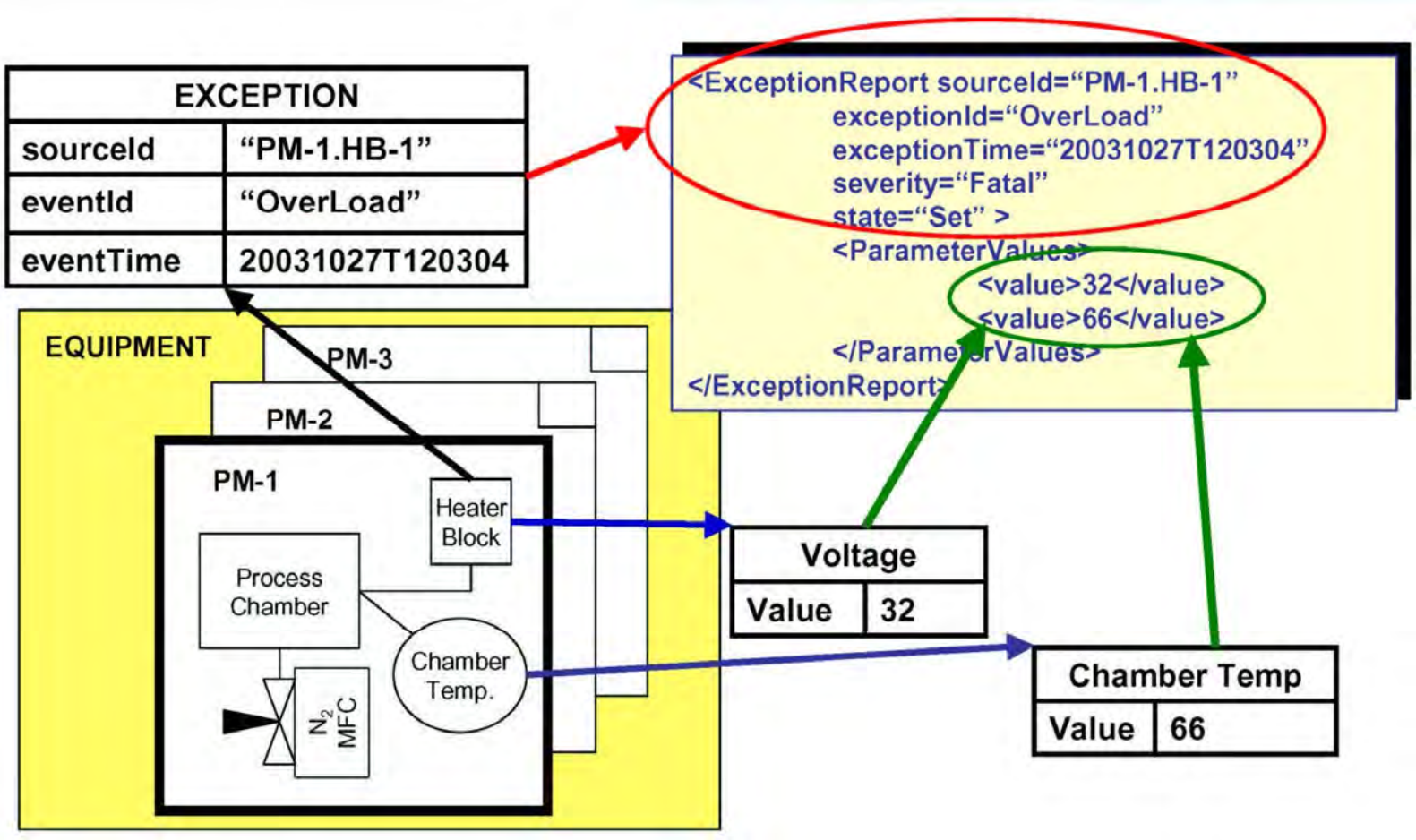


Exception Report Definition

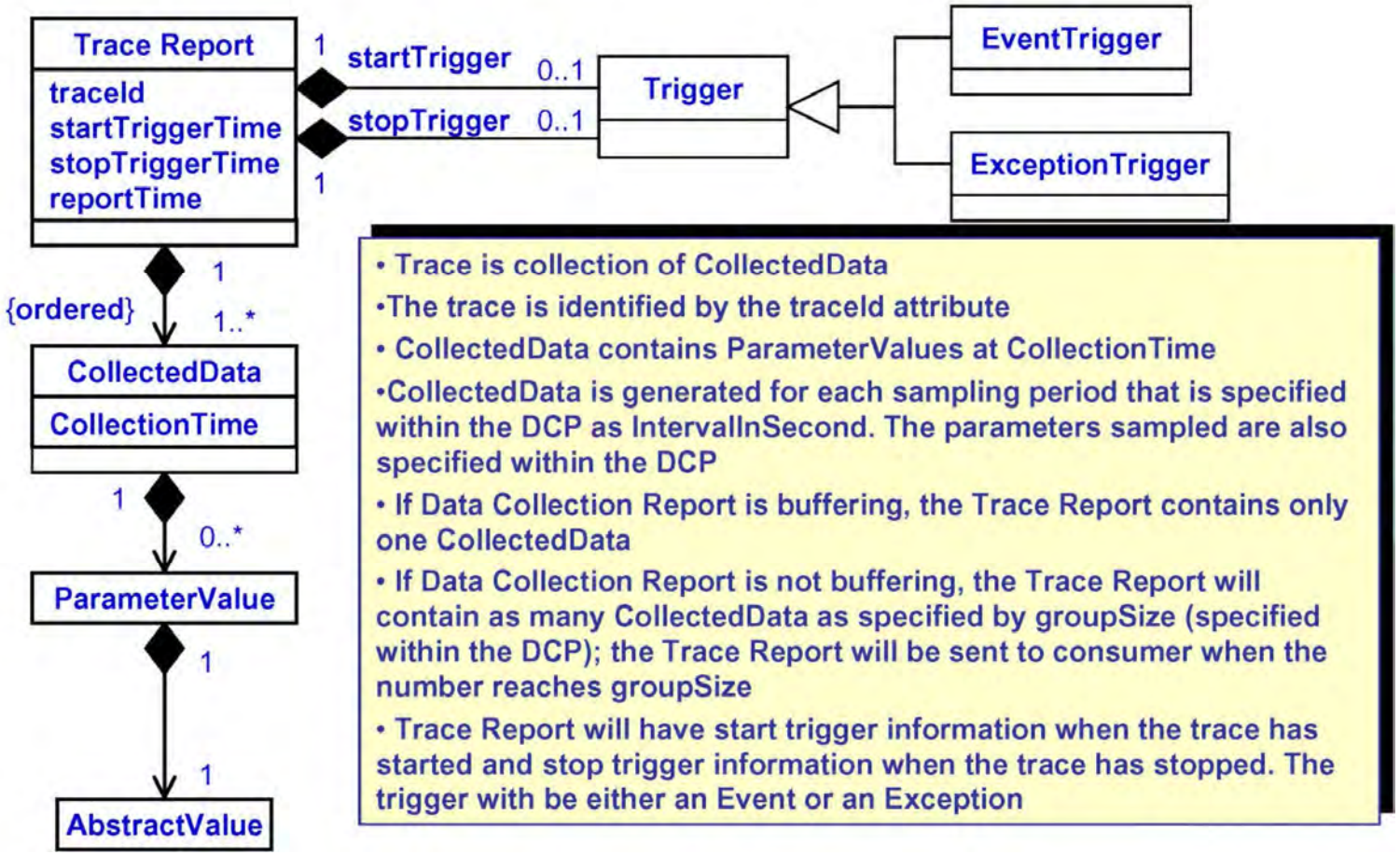


- An Exception Report is sent to a consumer when an exception occurs and the consumer requested the Exception Report
- "sourceId", "exceptionId", "state" and "severity" are used to identify the exception
- If the exception has state, then the information of SET or CLEAR must be set
- An Exception Report may have zero or more ParameterValues
- The ParameterValues are listed in E125
- ParameterValue provides the actual data value
- AbstractValue represents that here may be a value in a data type; the formats are different in data types

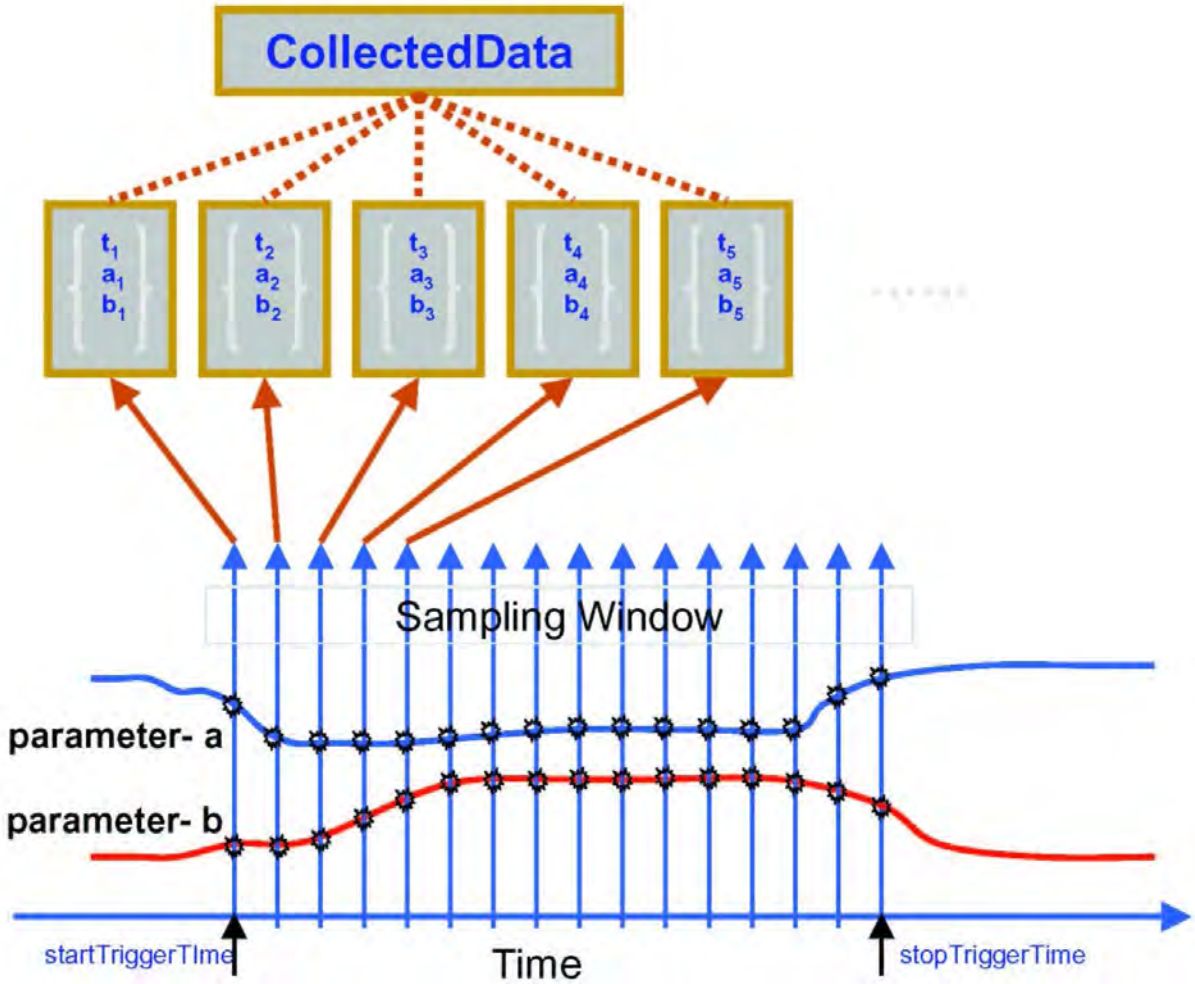
Example for Exception Report



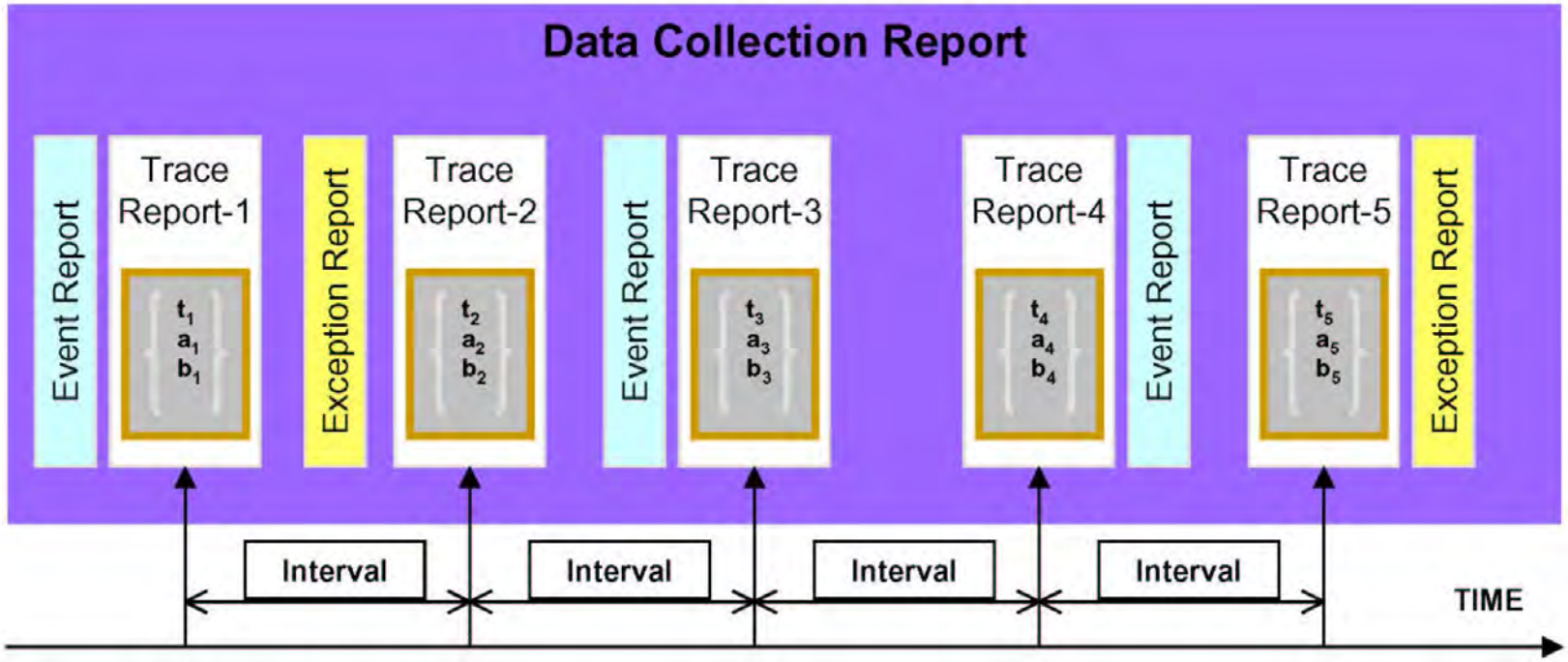
Trace Report Definition



Example of Collection Data

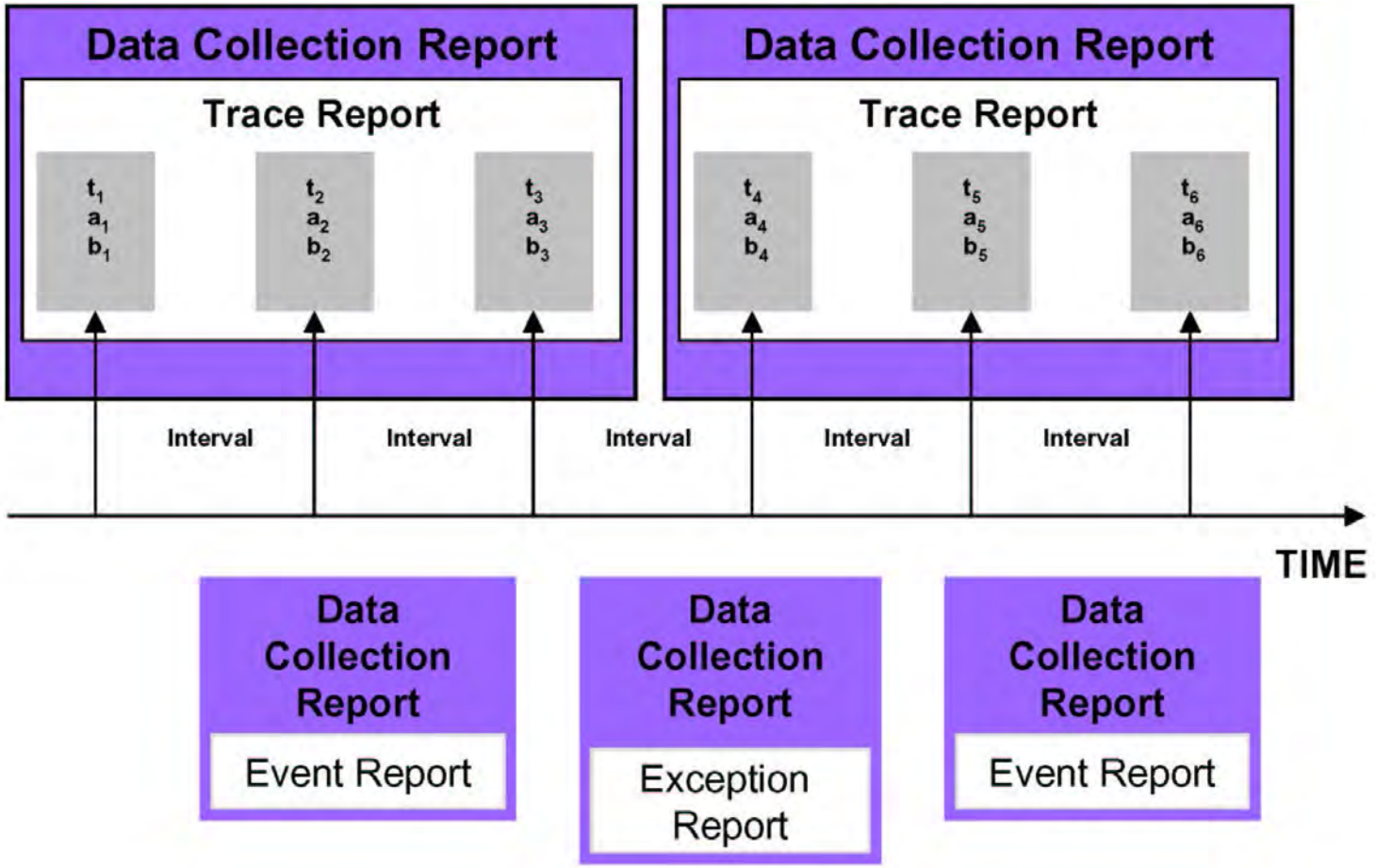


Example of Trace Report with DCR Buffering



All reports of Event, Exception and Trace are mixed in the Data Collection Plan

Example of Trace Report without DCR Buffering (groupSize=3)



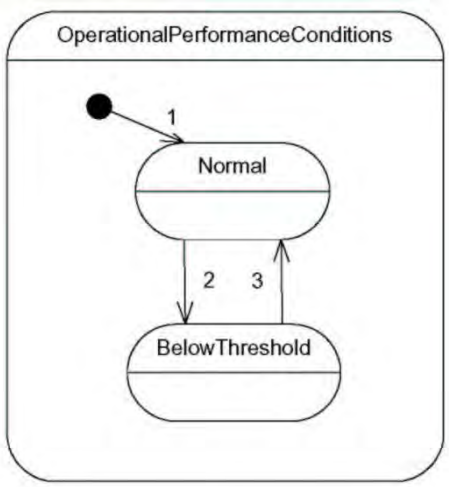
Report is sent in the Data Collection Report for each one

Message Trace Report Example

```
<Trace traceId="APC7400" startTriggerTime="20030925T160000"      stopTriggerTime="20030925T160100"
reportTime="20030925T160110">
        <Trigger>
                <StartTrigger>
                        <EventTrigger sourceId="PM-1" eventId="ProcessStart"/>
                </StartTrigger>
        </Trigger>
        <TraceResults>
                <Data collectionTime="20030925T160000"
                        <ParameterValues>
                                <PV>1234</PV>
                                <PV>1234</PV>
                        </ParameterValue>
                </Data>
                <Data collectionTime="20030925T160030"
                        <ParameterValues>
                                <PV>1234</PV>
                                <PV>1234</PV>
                        </ParameterValue>
                </Data>

        </ TraceResults >
</Trace>
```

E134 Equipment Performance Self-Monitoring



- **Allows equipment to warn clients of performance problems**
  - **Notifies consumers when performance is below the threshold**
- **Suppliers define performance thresholds**
  - **Supplier defines criteria and mechanism/algorithm for detecting performance related problems**
- **Decision regarding how to respond is owned by the factory**
  - **May decide to disable some DCP's or continue running, depending on factory policy**
  - **Equipment may terminate all DCP activity if the equipment/material/people is in danger**

Summary

- E120 provides a consistent vocabulary for describing equipment structure
- E125 provides a comprehensive set of data structures to describe information available from the equipment
- E125 and E120 together simplify the naming and identification of equipment data items
- E125 and E134 together create a more manageable and flexible data collection environment