Smarter Manufacturing with SEMI Standards: Practical Approaches for Plug-and-Play Application Integration

17th European Advanced Process Control
And Manufacturing (APC|M) Conference
Dublin, Ireland April 10-12, 2017

Alan Weber – Cimetrix Incorporated

Outline

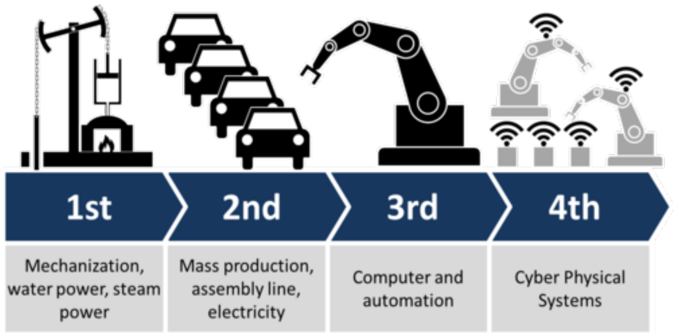
- What is “Smart Manufacturing?”
- Importance of equipment models
- SEMI Standards evolution
- Thoughts on “plug and play”
- Factory application examples
- Conclusions

What is "Smart Manufacturing?"

From Industry 4.0 Wikipedia…

- "… cyber-physical systems monitor physical processes, create a virtual copy of the physical world and make decentralized decisions.

- Over the Internet of Things, cyber-physical systems communicate and cooperate with each other and with humans in real time…"

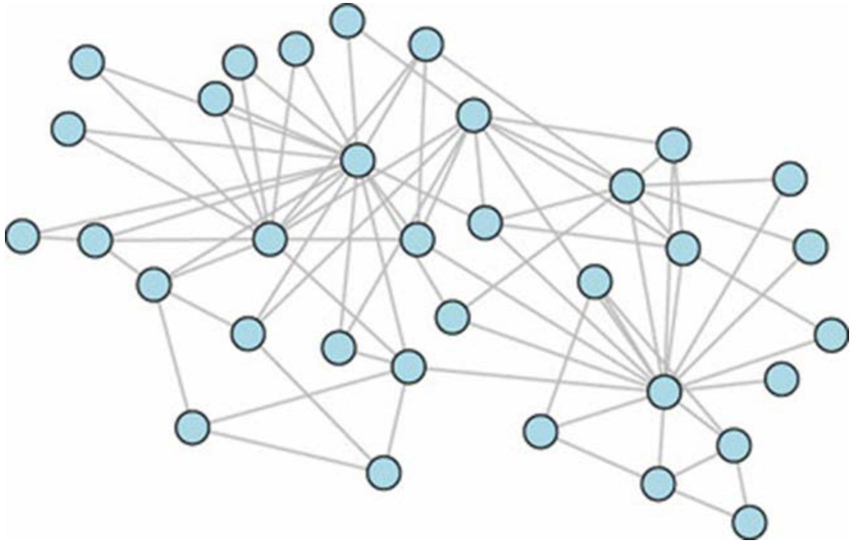


cimetrix®

Components of a Smart Factory

Attributes of all these connected "things"

- Discoverable
- Autonomous
- Model-based
- Secure
- Self-monitoring
- Easy to use
- Compact
- Standards-based



Imagine the collaborative behavior that could emerge !

Importance of equipment models

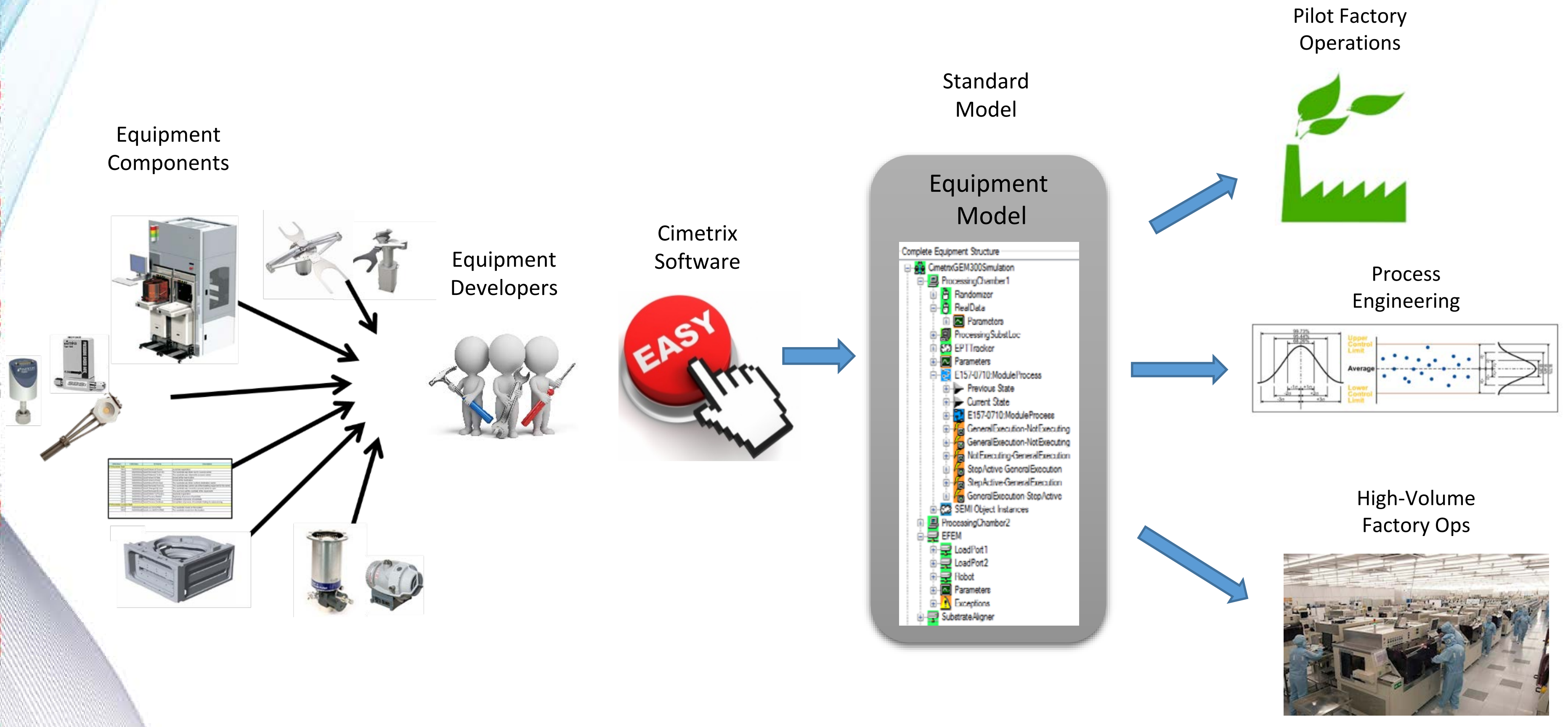
Most important Smart Factory data source

- Equipment models are useful
  - Help understand equipment and process behavior
  - Improve communication with suppliers
- Explicit, standard models are especially useful
  - Support generic applications across equipment types
  - Enable performance benchmarking within/among fabs
- Events and associated data offer untapped benefit
  - Time lost can never be recovered
  - Time is the ultimate unifying concept
- Modeling principles developed for the EDA standards apply equally well to GEM equipment

cimetrix®

The equipment model value chain

Control	Connect	Collaborate	Visualize	Analyze	Optimize



<u>KPIs (metrics)</u>

- Time to money
- Yield
- Productivity
- Throughput
- Cycle time
- Capacity
- Scrap rate
- EHS

cimetrix®

Evolution of equipment models

Referenced in SEMI standards

Natural language analogy…

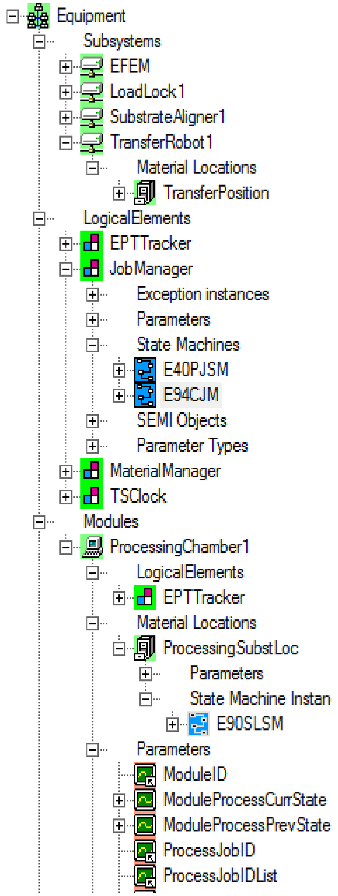
- SECS-I vocabulary (data items)
- SECS-II grammar (streams/functions)
- GEM sentences (capabilities)
- GEM300 conversations (scenarios)
- EDA improv theatre (dynamic DCPs)
- E164 improv theatre with a point (common model)
- <tbd> spontaneous flash mob (IIoT, …)

cimetrix®

Model structure examples

The __good__, the bad, and the ugly

- Structure matches tool organization
- Clear naming convention for all elements
- Complete coverage of important tool behaviors
- Consistent with applicable industry standards



Model structure examples

The good, the bad, and the ugly

- Structure is completely flat or excessively deep
- Opaque or inconsistent naming conventions
- Incomplete/poor coverage of important tool behaviors
- Partial implementation of industry standards

Tool

SV	DV	CE	EC
SV1	DV1	CE1	EC1
SV2	DV2	CE2	EC2
SV3	DV3	CE3	EC3
SV4	DV4	CE4	EC4
SV5	DV5	CE5	EC5
SV6	DV6	CE6	EC6
SV7	DV7	CE7	EC7
SV8	DV8	CE8	EC8
SV9		CE9	EC9
SV10		CE10	EC10
SV11		CE11	EC11
SV12		CE12	EC12
SV13		CE13	EC13
SV14		CE14	EC14
SV15		CE15	
SV16		CE16	
SV17		CE17	
SV18		CE18	
SV19			
SV20			
SV21			
SV22			
SV23			

cimetrix®

Model structure examples

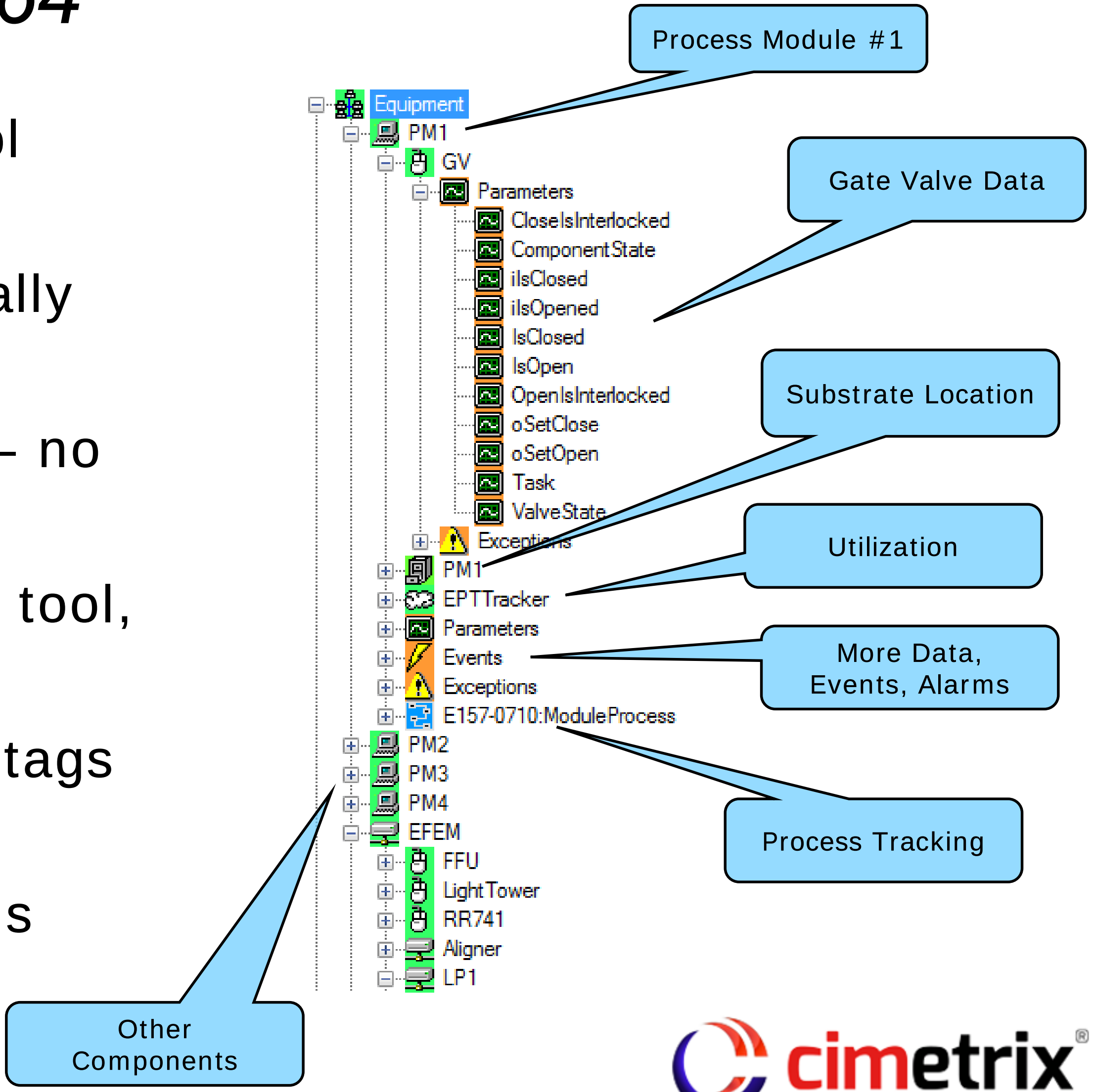
The good, the bad, and the ***ugly***

- No apparent structure, no resemblance to tool
- No apparent naming convention
- Indeterminate coverage of tool behavior
- Haphazard implementation of industry standards

Standard metadata model benefits

First specified by SEMI E164

- Model structure exactly reflects tool hardware organization
- Complete description of all potentially useful information in the tool
- Always accurate, always available – no additional documentation required
- Common point of reference among tool, process, and factory stakeholders
- Source of unambiguous identifiers/tags for database [auto] configuration
- Enables "plug and play" applications



cimetrix®

SEDD – SECS Equipment Data Dictionary

SEMI E172 highlights

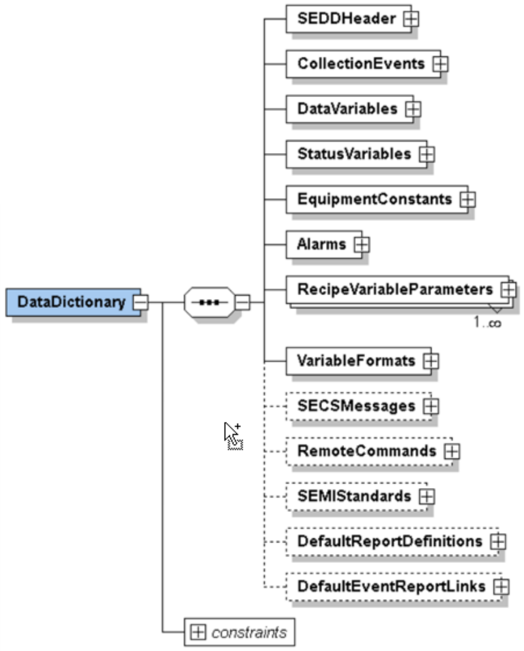
- Purpose
  - Describe an equipment's SECS interface in a format that is both human- and computer-readable
  - Replace vendor-specific electronic formats (typically multi-tab Excel spreadsheets at best) with true industry standard
- Benefits
  - Fabs can streamline equipment integration software development
  - OEMs can adopt standard rather than creating their own
  - Use broadly available software tools to create and manage

E172 is for GEM what
E120/E125 is for EDA/Interface A !

cimetrix®

SEDD – SECS Equipment Data Dictionary

Schema and examples



	Collection Event	Data Variable	Status Value	Equipment Constant	Alarm	RVP
1	CEID	VID	SVID	ECID	ALCD	RVPName
2	CENAME	DVVALNAME	SVNAME	ECNAME	ALID	MinValue
3	VALIDDVS	UNITS	UNITS	ECMIN	ALTX	MaxValue
4	RelevantVariables	Format	Format	ECMAX	SetEvent	Description
5	StateTransition	MinValue	MinValue	ECDEF	ClearEvent	Source
6	Description	MaxValue	MaxValue	UNITS	Description	Format
7	Source	Description	Description	Format	Source	Units
8	SEMIStandard	Source	Source	Description	SEMIStandard	
9		SEMIStandard	SEMIStandard	Source		
10				SEMIStandard		

....

```
<CollectionEvent>
    <CEID>2000</CEID>
    <CENAME>Control:OffLine</CENAME>
    <VALIDDVS xsi:nil="true"/>
     <RelevantVariables xsi:nil="true"/>
    <StateTransition>
        <StateModel>E30-0307:Control</StateModel>
        <TransitionID/>
        <PreviousState/>
        <NewState>Off-Line</NewState>
    </StateTransition>
    <Description>Equipment control status is now OffLine</Descripti
    <Source>TrackSys404</Source>
    <SEMIStandard>E30-0307</SEMIStandard>
</CollectionEvent>
```

```
<Alarm>
    <ALCD>0</ALCD>
    <ALID>3033</ALID>
    <ALTX>Carrier Placement Error</ALTX>
    <SetEvent>4066</SetEvent>
    <ClearEvent>4067</ClearEvent>
    <Description>Carrier placement on port is incorrect</Description>
    <Source>TrackSys404/EFEM/LoadPort1</Source>
    <SEMIStandard>E87-0709</SEMIStandard>
</Alarm>
```

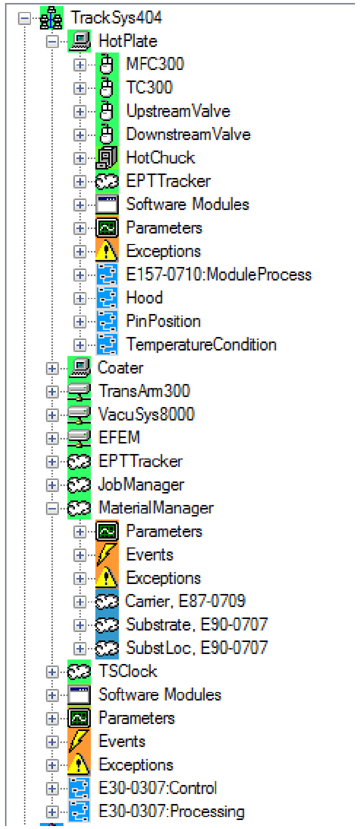
Equipment model structure

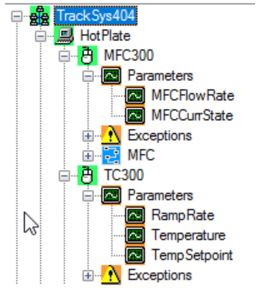
Embedded in <source> elements

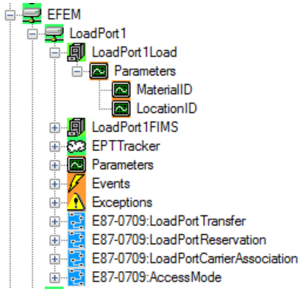
```xml
<CollectionEvent>
    <CEID>2046</CEID>
    <CENAME>MFC:NoState-MFCOff</CENAME>
    <VALIDDVS>
        <VID>0</VID>
    </VALIDDVS>
    <StateTransition>
        <StateModel>MFC</StateModel>
        <TransitionID/>
        <PreviousState>NoState</PreviousState>
        <NewState>MFCOff</NewState>
    </StateTransition>
    <Description>MFC is initialized</Description>
    <Source>TrackSys404/HotPlate/MFC300</Source>
</CollectionEvent>
```

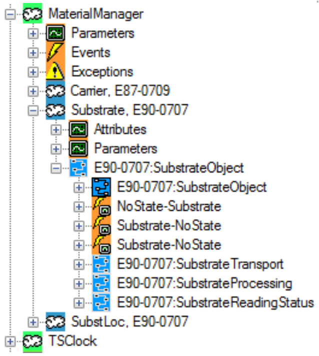
```xml
<StatusVariable>
    <SVID>7134</SVID>
    <SVNAME>PortTransferState1</SVNAME>
    <Format>PortTransferStateEnum</Format>
    <Description>Current transfer state of this load port.
    unsigned integer (SECS-II code 51). The enumeration val
    E87.1-0709 Table 4.</Description>
    <Source>TrackSys404/EFEM/LoadPort1</Source>
    <SEMIStandard>E87-0709</SEMIStandard>
</StatusVariable>
```

```xml
    <StateTransition>
        <StateModel>E90-0707:SubstrateProcessing</StateModel>
        <TransitionID>11</TransitionID>
        <PreviousState>NeedsProcessing</PreviousState>
        <NewState>InProcess</NewState>
    </StateTransition>
    <Description>SubstrateID specified by variable; Substrate starts
    <Source>TrackSys404</Source>
    <SEMIStandard>E90-0707</SEMIStandard>
</CollectionEvent>
```









cimetrix®

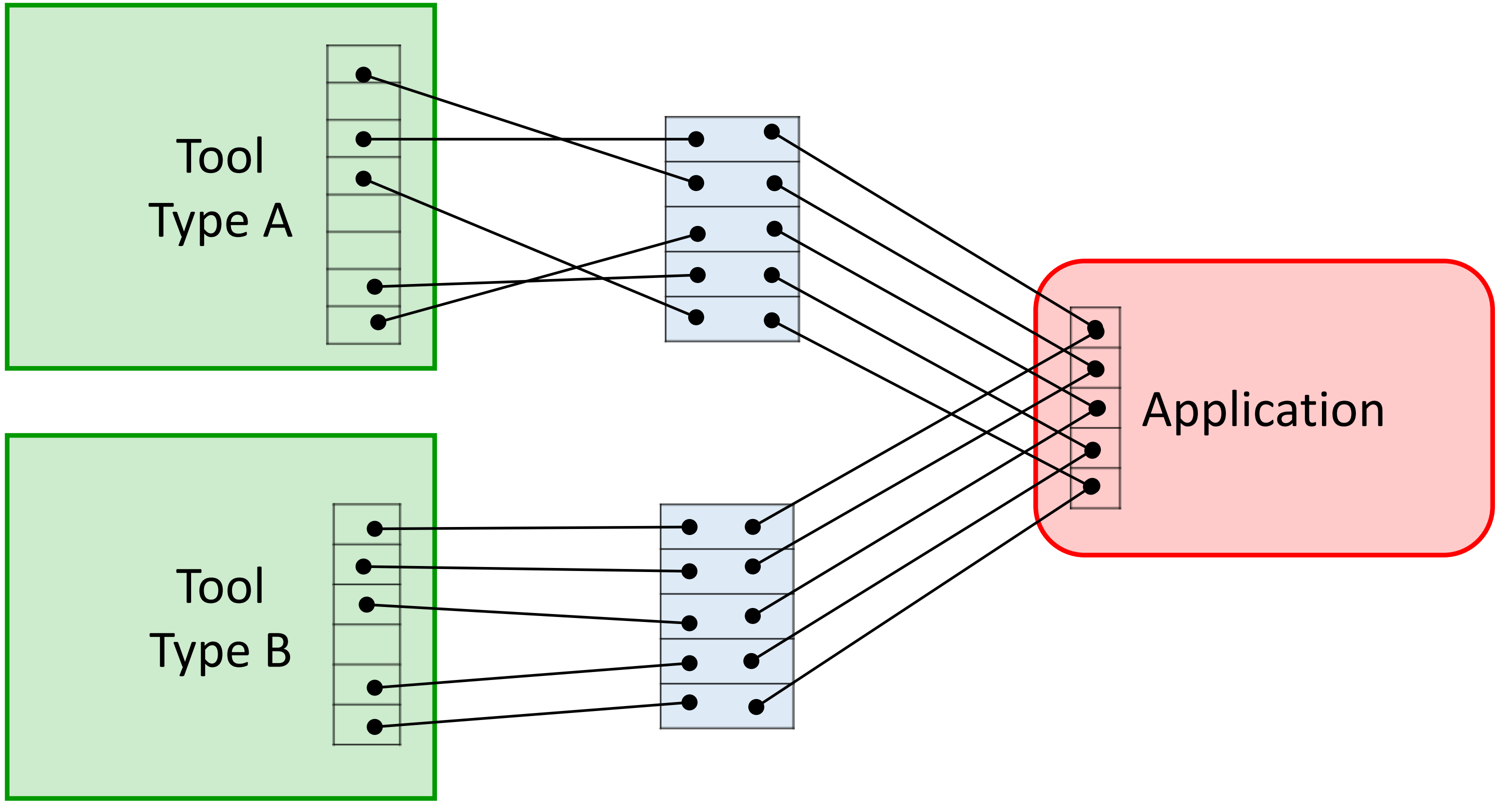
Thoughts on “plug and play”

In the context of Smart Manufacturing

- What is it?
  - Application integration concept
  - Equipment directly supports application capability without additional programming or configuration
  - Applications can make assumptions about semantics of tool data (i.e., about what things mean)
- Who benefits and how?
  - Factory customers greatly reduce application development/integration cost and time to market because of commonality across equipment types

Achieving application "plug and play"

Using "connectors" to map SEDD model elements



Plug and play application integration
Example tool data required (1)

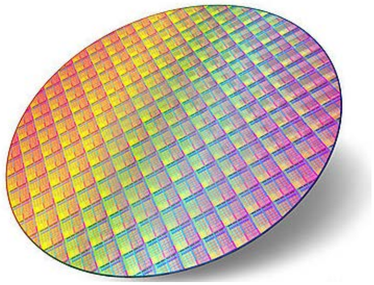
Application	Tool Data Required
OEE (Overall Equipment Effectiveness)	Transition events and status codes sufficient to classify equipment states for all time periods
Intra-tool material (substrate) tracking	Material tracking events; material location state indicators and state change events
Process execution tracking	Start/stop events for all processing modules; recipe step indicators and step change events for all processing modules that support multi-step recipes
WTW (wait time waste) analysis (aka Product Time Measurement)	The combination of events required for the intra-tool material flow and process execution tracking applications (see above) and context data required to classify material states for all time periods (see the SEMI E168 Product Time Management standard for a deeper explanation)
Time-based PM (Preventative Maintenance)	Run timers at the FRU (field replaceable unit) level
Usage-based PM	Usage parameters and accumulators appropriate for each FRU, such as time-in-state, execution cycles, fluid flow rates, consumables flow rates, power consumption, etc.
Condition-based PM	Meaningful “health indicators” for each FRU
FDC (Fault Detection and Classification)	Equipment/process parameters required by specific fault models and associated context information (this is difficult to do completely because most FDC models are “trained” with knowledge of “good” and “bad” runs, which is not known to the OEM a priori)

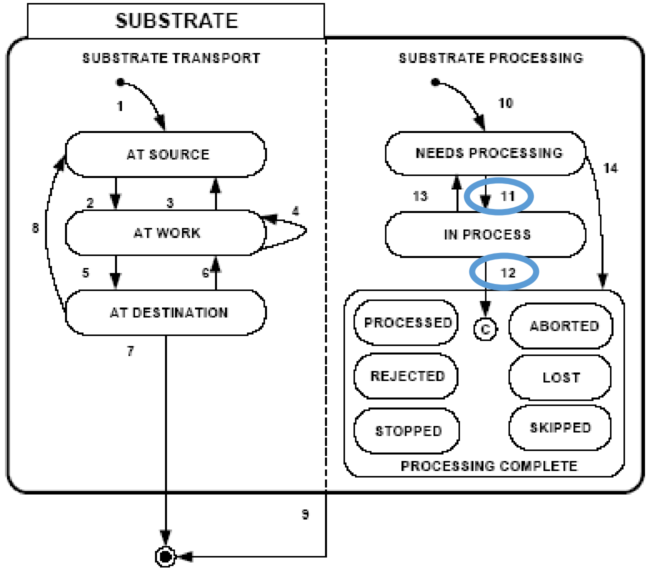
Plug and play application integration
Example tool data required (2)

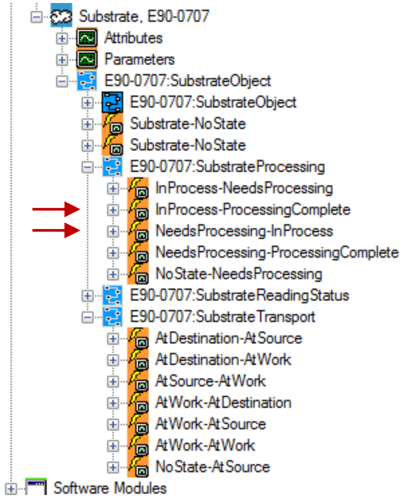
Application	Tool Data Required
Equipment configuration monitoring	Vector of important equipment constants with expected values and acceptable ranges; may need to support multiple sets, if the values are setup-dependent. Designed to catch human error resulting from operator manual adjustments.
Component fingerprinting	Performance parameters for key equipment mechanisms, including command/response signals at the sensor/actuator level
Static job scheduling	Setup and execution times per product/recipe combination and current setup information
Real-time job dispatching	Estimate of current job completion time; estimate of completion time for all material queued at the equipment
Factory cycle time optimization	Material buffer contents, job queue information
Real-time dashboard	Equipment/component production status indicators
Equipment failure analysis	Meaningful alarm/fault codes and recent history/statistics
Remote recipe selection	Recipe directory with meaningful header information (version #, checksum, etc.) to support reliable selection
Recipe verification	Checksum calculation spec to compare equipment-resident recipe with host version
Run-to-run control	Identification of recipe adjustable parameters and commands to remotely update them

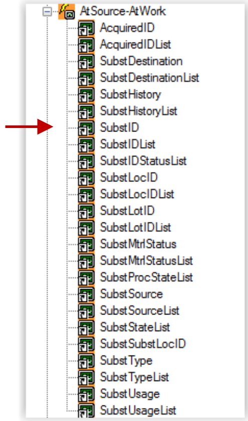
Substrate tracking

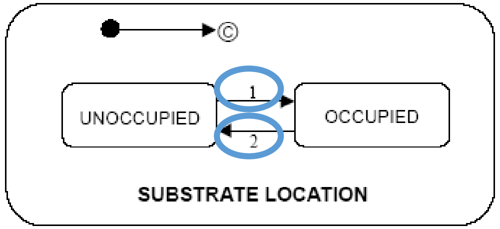
SEMI E90 state machines and model content

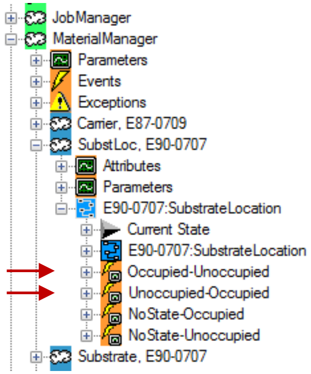


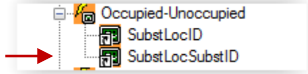








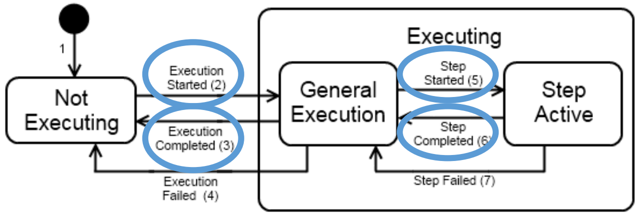


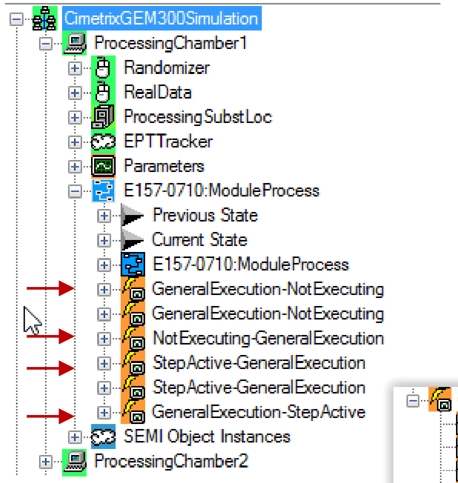


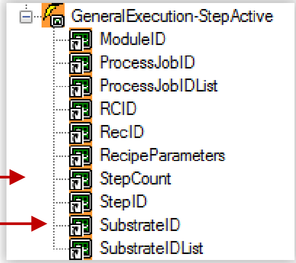
cimetrix®

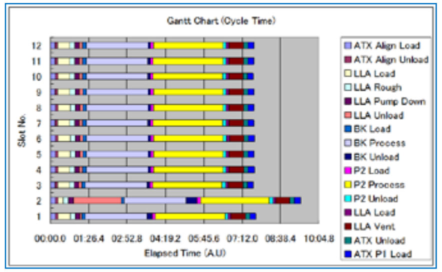
Process execution tracking

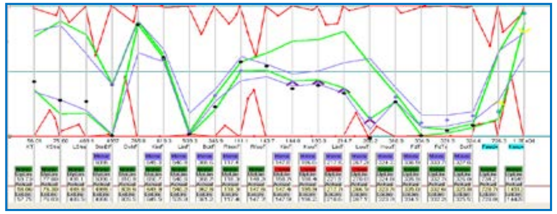
E157 state machine, model content, and results











cimetrix®

Conclusions

- Models help components of a Smart Manufacturing system understand one another
- Models have been at the core of SEMI's Information and Control standards for decades
- The sophistication of these models is now sufficient to support “plug and play” application integration
- Industry standard models can greatly reduce factory application development/integration cost
- Models will play an increasingly important role as the number and variety of components multiply