EEC High-Level Requirements for Advanced Process Control

15 June 2004

Equipment Engineering Capabilities Data Taxonomy working group
International SEMATECH

Co-Chairs:
David Busing
Karl Gartland

Members:

Keith Bennet
Ray Bunkofske
Gino Crispieri
Roger Eastvold
Rudi Hendel
Leonard Kamlet
Shige Kobayashi
James W Martin
James Moyne
John Pace
Ray Peterson
Lance Rist
Scott Smith
Jeff Toth
Brad Van Eck

Revisions:

Date	Description	Editor
3/15/02	Initial draft posted to International Sematech website	David Busing
3/27/02	Copy editing	Harvey Wohlwend
4/15/04	Clarify Parametric Data Sample Period and Throughput section and Latency and Buffering section	James Martin, Dave Busing, Karl Gartland, Ray Bunkofske, Harvey Wohlwend

Executive Summary

This document presents an exposition of general capabilities, data taxonomy, and requirements for Advanced Process Control, referring to Fault Detection and Classification (FDC) and Run-To-Run (R2R) control. The purpose of this exposition is to determine the impact of supporting APC via an Equipment Engineering Data Interface, and to aid in the design thereof. At the time of this writing, the SEMI Diagnostic Data Acquisition Interface under development is the leading proposal for an EEDI.

This document focuses on the data acquisition from the tool for use cases outside the tool. It does not address specific architecture, protocols, data message formats, or security capabilities. Because the demand for data capability has consistently outpaced the capability delivered by past specifications, some of the statements herein are speculative. Furthermore, due to the variety of approaches to APC within the industry, the nature of the consensus in this document is a combination of (1) requirements for specifying deliverable system functionality, and (2) abstract definitions for communication purposes between suppliers and users.

The technology used for data transfer must support (1) at least 1000 independent name-value pair messages, each at a sample period of one second, *OR* (2) at least one trace message of 1000 variables at sample period of 100 milliseconds. In either case, the latency from physical event to possible transmission out the data port of the tool can be no larger than 100 milliseconds. The tool is required to buffer up to a maximum of 100 seconds of data for APC usage according to the two data options above before permitting any drop out of data.

This document represents the completion of the first phase of the EECDT effort that has been focussed solely on APC requirements. In the next phase, the EECDT working group will expand its scope to address:

- tool-specific and application-specific support requirements for APC, including the special case of integrated metrology.
- EEC data considerations above the tool interface, e.g., application-to-application, MES-to-application, etc.
- EEC data capabilities not covered under e-Diagnostics or APC, if any.

Ultimately, this working group intends to consolidate data taxonomy, data capabilities, and data requirements from all known sources, including the SEMI DDA task force and International SEMATECH's e-Diagnostic/EEC initiative, into one comprehensive master document.

Table of Contents

1 Introduction / Purpose 4
2 Terminology 4
2.1 Reference Standards 5
3 General APC Capabilities and Requirements 6
3.1 General Restrictions 6
3.2 Data Transfer From Equipment 6
3.3 Parametric Data Performance 7
3.3.1 Parametric Data Resolution and Accuracy 7
3.3.2 Parametric Data Sample Period and Throughput 8
3.3.3 Latency and Buffering 8
3.3.4 Sampling Period Uniformity and Skew 9
3.4 Bandwidth Management 10
3.4.1 Dynamic Data Collection 10
3.4.2 Data Customization 10
3.4.3 Data Compression 11
3.5 Interdiction 11
Data Context for APC 13
3.6 Specific Context Requirements for FDC 13
3.7 Specific Context Requirements for Run to Run Control 14
3.8 Dynamic Data Context 15
3.8.1 Timestamp 15
3.8.2 Tool Resources 15
3.8.3 Material Resources 16
3.8.4 Operations 16
3.9 Static Data Context 16
3.9.1 Parameter and Event Context 17
3.9.2 Data Refinement 17
3.9.3 Data Purpose 18
3.9.4 Discoverable Interface 19
4 Data Quality 20
4.1 Data Availability and Reliability 20
4.2 Sensor Tracking and Self Checking 20
4.3 Bandwidth Monitoring 20
Appendix A: APC “Learning” versus APC “Doing” 22
Appendix B: Features of Extracted Data 23

1 Introduction / Purpose

This document presents an exposition of general capabilities, data taxonomy, and requirements for Advanced Process Control, referring to Fault Detection and Classification (FDC) and Run-To-Run (R2R) control. The purpose of this exposition is to determine the impact of supporting APC via an Equipment Engineering Data Interface, and to aid in the design thereof. At the time of this writing, the SEMI Diagnostic Data Interface under development is the leading proposal for an EEDI.

This document focuses on the data acquisition from the tool for use cases outside the tool. It does not address specific architecture, protocols, data message formats, or security capabilities. Because the demand for data capability has consistently outpaced the capability delivered by past specifications, some of the statements herein are speculative. Furthermore, due to the variety of approaches to APC within the industry, the nature of the consensus in this document is a combination of (1) requirements for specifying deliverable system functionality, and (2) abstract definitions for communication purposes between suppliers and users.

2 Terminology

- **Backfill—**To substitute values in variables that change less frequently than the sample rate of a trace containing those variables; the values that have not changed are backfilled with their previous values.
- **Compression (Data Compression)—**Data volume is reduced in a manner that is intended to preserve and not limit any applications of the data. For a non-destructive compression method, all details of the original uncompressed data set may be *recreated* from the compressed data set, e.g., running delta trace, etc. A compression method for which complete recreation is not possible is destructive.
- **Control Event—**A Boolean static property of an event that may affect the accuracy of its timing. It is mutually exclusive with Response Event. A control event is generated by the equipment control system to motivate an action, and is independent of the actual execution of the action.
- **Context—**Supplemental information that is used to qualify other data values, provides the means to interpret other data values, or describes a relationship between two or more data entities. This includes *static context data* (i.e., parameter name, parameter valid data range, parameter units) and *dynamic context data* (i.e., current control limits, current recipe step, current job, current wafer).
- **Drift—** A measure of the accuracy of sample timing uniformity. Drift is the slope of a linear trend in the actual inter-sample times.
- **Jitter—** A measure of the accuracy of sample timing uniformity. Jitter is conventionally measured as the difference in period from cycle-to-cycle.

- **Post-Mortem—**A non-technical qualitative term to describe a characteristic of a data or actuation system that makes that system only effective for applications after the fact, as opposed to real-time applications.
- **Real-Time—**A non-technical qualitative term to describe a characteristic of a data or actuation system that makes that system effective for interactive applications, as opposed to post-mortem applications.
- **Response Event—**A Boolean static property of an event that may affect the accuracy of its timing. It is mutually exclusive with Control Event. A response event is detected or measured by the equipment control system through any of its input systems, e.g., sensors, GUI, communication ports, etc.
- **Run**—a complete operation in a single processing station without a change of substrates.
- **Run-To-Run—**A run-to-run application customizes certain recipe parameters between runs to improve processing performance. Run-to-run control is limited to control that may be applied before beginning a run in response to either previous run results (feedback) or predictive data about the incoming material (feedforward). Adaptive or corrective control during a run, other than FDC, is not considered part of run-to-run control.
- **Sample Period**—The nominal time in-between two successive samples.
- **Sampling—**The capability of selecting the sample period, start trigger, and end trigger of a trace.
- **Selection**— The capability of selecting the parameters reported in a trace.
- **Shipping**—The capability of selecting data messages per unit time that is issued by a data reporting system.
- **Skew**— Skew is the difference in actual acquisition times between variables in the same sample. The overall skew for the sample is shown as the difference in acquisition times between the first and last parameter values acquired.
- **Zero-Order Fill**—A particular form of backfilling data, where the values in question have a zero, null, or "no state" value (depending on the variable) signifying that the variables are currently not applicable, e.g., out of scope.

2.1 Reference Standards

- SEMI-E40—Standard for Processing Management
- SEMI-E53—Event Reporting
- SEMI-E87—Provisional Specification for Carrier Management
- SEMI-E90—Specification for Substrate Tracking
- SEMI-E94—Provisional Specification for Control Job Management

3 General APC Capabilities and Requirements

Capabilities and requirements apply specifically to Advanced Process Control, referring to Fault Detection and Classification (FDC) and Run-To-Run (R2R) control. Requirements for non-APC applications are not represented. Emphasis is furthermore placed on applications for APC “doing” rather than APC “learning” (see discussion, Appendix A).

An FDC application collects data from a tool and analyzes that data for process-oriented faults. If a fault is detected, the FDC application issues a command to the tool to gracefully suspend processing at the related processing station.

A run-to-run application customizes certain recipe parameters between runs to improve processing performance. A run in this context applies to a complete operation in a single processing station without a change of substrates. Run-to-run control is limited to control that may be applied before beginning a run in response to either previous run results (feedback) or predictive data about the incoming material (feedforward). Adaptive or corrective control during a run, other than FDC, is not considered part of run-to-run control and is not addressed in this document.

The main requirements in this document pertain to the transfer of data from process equipment to a point outside of that process equipment. Overall APC systems may rely on other forms of data transfer not addressed in this document including data transfer *from* applications, inspection and metrology, or MES or central maintenance information systems.

3.1 General Restrictions

Equipment must satisfy the following general restrictions when implementing support for APC applications:

- APC data collection and/or control actions can never undermine safety.
- AEC/APC data collection and transmission must not inhibit, degrade, or usurp basic operational tool control, e.g., factory automation, host control.
- During operation of the tool for production, overall data collection and transmission must not adversely impact the performance of the tool with respect to throughput, cycle time, or OEE.

3.2 Data Transfer From Equipment

All data must be translatable and deducible without software customized for specific equipment and/or applications. Data structures must be universally interpretable by the equipment and receiver either by standard data type, self-description, or by reference, e.g., a configuration file, object library, etc.

APC data is predominantly trace data with characteristics and behavior similar to that specified in SEMI E53. APC also relies on certain event data described in section 4.

Trace data is comprised of variables sampled over time. For each timestamp, one or more variables of data are reported in a pre-defined vector or list. A series of such vectors are reported at intervals according to a predefined sample period. The act of trace data collection for any trace begins with a start trigger event and continues until a finite number of samples have been gathered, a fixed amount of time has passed, and a stop trigger event has occurred.

Within a trace, data should be reported in standard engineering units, e.g., volts, cm, torr, etc., and conventional orders of magnitude, e.g., do not report the value "1026" to mean "10.26".

For members of a vector whose values change less frequently than the sample rate of the vector itself, the values that have not changed must be backfilled with their previous values. Transient variables must be backfilled with a zero, null, or "no state" value, signifying when the variables are currently out of scope.

3.3 Parametric Data Performance

Parametric data collection must meet several performance requirements presented in this section.

3.3.1 Parametric Data Resolution and Accuracy

Parametric data must have sufficient resolution and accuracy in order to distinguish signal noise from potentially subtle trends in the data. Typical order of magnitude of process tolerances are in the single digits: less than 10%. Assuming 1% process sensitivity, a resolution one order of magnitude lower is required, or 0.1%.

Data must be provided with a resolution of no less that 0.1% of full scale or 0.1% of normal operating range, whichever is finer resolution. For example, for a sensor that has a full scale operating range between 0 and 12 volts, a resolution of at least 0.012 volts is needed. If the normal operating range is between 9 and 11 volts (a 2 volt range), a resolution of 0.002 volts is needed.

A resolution of 0.1% requires an absolute accuracy of 0.05% is required for the least significant bit of the data to have meaning. Data shall be provided with an absolute accuracy of no less than .05% of full scale or .05% of normal operating range, whichever is finer resolution.

3.3.2 Parametric Data Sample Period and Throughput

Parametric data must have a sufficient resolution across time to distinguish signal characteristics that occur at a high speed. Also, complex analyses may draw upon many variables. The technology used for data transfer must support a maximum throughput of 10,000 data values per second. On-tool sampling rate requirements are process dependent. Off-tool transmission rate requirements are application dependent. This does not preclude additional capabilities that would be required for e-Diagnostics or other use cases.

The typical tool should provide 50 variables per chamber at a maximum on-tool sample rate of 10Hz. For some specialized processes such as rapid thermal and flash anneal, an additional smaller number of (up to, for example, a total of 30-40) critical variables may require a maximum on-tool sample rate of 200Hz.

The actual data content and required number of simultaneous traces may vary depending on tool type and/or process stations. For key process stations, such as chambers in an RF plasma etching system, it is highly recommended that each process station support its own independent trace of at least 50 parameters with maximum sample periods of 100ms.

It is recommended that the minimum sample period for parameters be at 1% or less of the duration of a related recipe step duration. An example would be 100 ms for a 10 second recipe step. The sampling period should also be at least twice as fast as the duration of the fastest occurring fault. However, sample periods shorter than 100ms are not expected to be common.

3.3.3 Latency and Buffering

In general, data must be transmitted as soon as possible in order to support real-time analyses and to support timely interdiction where it is warranted.

The number of data samples contained in each independent off-tool transmission shall be constant, and conform to on-tool buffering parameters that are configured by the equipment user. Off-tool transmission latencies are acceptable up to 1 second following the completion of the specified number of samples within the specified on-tool buffering period. It is acceptable for equipment to restrict off-tool transmission frequencies to maximum rates that are not identical to the on-tool sampling rates described in section 3.3.2.

For discrete event data, the latency from the occurrence of a physical event to possible transmission out the data port of the tool should be no larger than 100 milliseconds. “Possible transmission” allows for the situation that either the network or the client application may be unavailable to receive the data. As a result of such outages, data may (but is not required to) be buffered at the tool.

3.3.4 Sampling Period Uniformity and Skew

Trace data must be received at a sufficiently uniform sampling rate. In the absence of such uniformity, additional pre-processing of data is required in order to use many conventional time-series analyses; generally this process is an interpolation of the desired spacing of data points based on existing data points.

Figure 1 shows the nominal sample times for the first four samples of a trace. *Jitter* is conventionally measured from cycle-to-cycle and has more to do with precision. In the above example, one instance of jitter may be calculated as |(t1 – t0) – (t2-t1)|, where t0, t1,t2, … are successive actual sample times. *Drift* is the slope of a linear trend in the actual sample inter-arrival times.

Skew is the difference in actual acquisition times between variables in the same sample. The overall skew for the sample is shown as the difference in acquisition times between the first and last parameter values acquired. To the extent that the sampling routine cycle is constant and predictable a phase offset may be applied as an effective correction.

For each sample, the time at which data is actually acquired must be within plus or minus 5% of the sample period, SP, of the nominal sample time. This specification simultaneously embodies notions of jitter, drift, and skew.

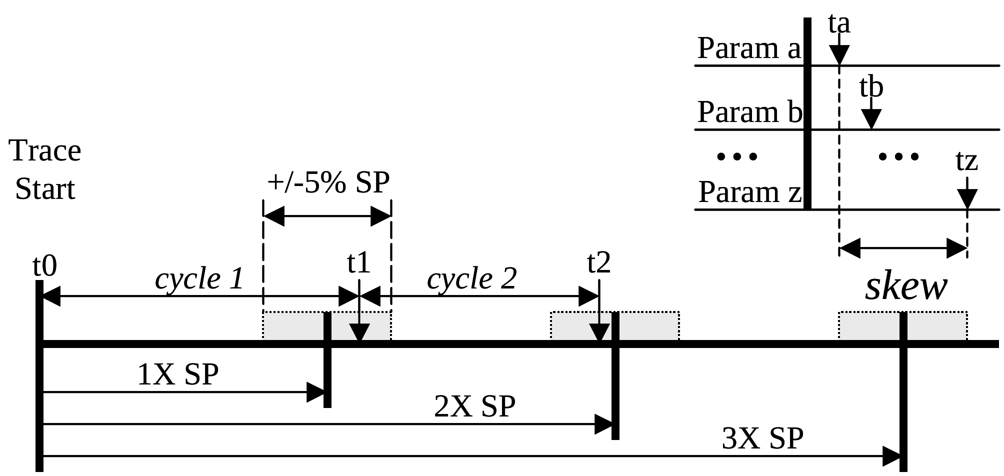

Figure 1: Sampling Uniformity of Trace Data

3.4 Bandwidth Management

It is not believed to be practical for tools to send all of their available parameters at the highest sample rates possible all the time. In order to manage the tool's limited processing and transmission resources, these resources must be managed in a manner appropriate for each tool type and potential set of APC applications. Tools for bandwidth management include dynamic data collection, customization, and compression.

3.4.1 Dynamic Data Collection

Dynamic data collection is the capability of the collection process to change from moment to moment. In terms of traces, this is the ability to start and stop traces at times of interest. This includes:

- starting traces on command, as is currently done through conventional SECS-II.
- starting traces by event triggers similar to behavior specified in SEMI-E53. Upon these events, data collection can be started or ended.

By starting and stopping different traces upon various events, the overall set of parameters sampled and the sample rate are changed.

Many equipment failures are related to changes in the transient characteristics of control systems. Because most control system transients happen at the boundaries of recipe steps, capturing data at recipe step boundaries is required. Furthermore, because the actions may vary significantly from step-to-step, e.g., gas flows, voltages, position, etc., the data of interest is also likely to change significantly.

Therefore, equipment is expected to support dynamic collection at least at the level of a recipe step, triggering at recipe step boundaries.

Data collection must have the ability to trigger based on a command from the host, specified equipment events, or a combination of a specified equipment event and status known to the equipment. As an example of the last case, if a wafer is present in a chamber and the chamber temperature exceeds a pre-defined threshold, then a trace may be triggered. If no wafer is present, the trace may not be triggered regardless of the chamber temperature.

3.4.2 Data Customization

Data customization is the capability allowing IC makers to tailor data collection to their specific applications. Customization is conventionally facilitated through a reasonably convenient automated interface at the equipment, such as the data customization capability of SECS-II.

Customization is related to but distinct from dynamic data collection. For example, a trace that is predefined by the supplier may be dynamic without being customizable by the IC maker. Furthermore, if the IC maker is offered a one-time choice of parameters, but not a choice of event triggers, data may be customizable without being dynamic.

Equipment must provide at least (1) the equivalent to SECS-II customization capability for selection and tagging, or (2) assurance that the IC maker's required data is contained within a fixed bulk or streaming transmission of data.

There are four aspects by which data may be customized:

- Selection—selection of parameter members in a trace vector.
- Sampling—For each trace, to select the sample period, start triggers, and end triggers.
- Tagging—The association of context information with primary data for the enrichment of subsequent analysis. This is discussed in section 4.3.
- Data shipping—The rate at which the accumulated samples are transmitted from the tool. This rate can be less than the sampling rate and can be configured to balance message overhead against message response time.
- Refining—transforming data for greater usefulness. Specific types are discussed in section 4.4.2.

Customizing the refining of data on the tool is ***not*** a requirement. This functionality, if any, is expected to be fixed for each tool and not customizable.

3.4.3 Data Compression

Current experience with data collection on a tool level indicates a serious problem with the ability of data acquisition hardware to keep up with the data generated by equipment. Data compression reduces data volume in a manner that is intended to preserve and not limit any applications of the data. For a non-destructive compression method, all details of the original uncompressed data set may be *recreated* from the compressed data set, e.g., running delta trace, etc. A compression method for which complete recreation is not possible is destructive.

It is understood that some but not all sensors require this compression, depending on the volume of data being transmitted by the sensor. Any data compression solution must not impact the sampling rate and data resolution requirements. Data shall be provided in compressed form. It is highly recommended that data compression should also be customizable by the IC maker.

3.5 Interdiction

To enact advanced process control, equipment must support points of interdiction where the MES or station controller may exert influence on the equipment's behavior. The EEDI is not the predicted interface for providing corrections or control to the tool during productive operation.

Interdiction requirements for FDC are limited to a pause, stop, or abort command that gracefully suspends any process station as soon as safely possible. Interdiction requirements for R2R are limited to setting parameters at the beginning of a run, and not during a run. Interdiction requirements shall be compliant with SEMI-E40.

Data Context for APC

Data context is supplemental information that is used to qualify other data values, provides the means to interpret other data values, or describes a relationship between two or more data entities. This section presents specific requirements for FDC and R2R, general requirements for dynamic context data (i.e., current control limits, current recipe step, current job, current wafer), general requirements for static context data (i.e., parameter name, parameter valid data range, parameter units), and some concepts and requirements related to a discoverable interface.

3.6 Specific Context Requirements for FDC

The event messages in Table 1 must be provided by the tool immediately prior to start of a run. A run in this context applies to a complete operation in a single processing station without a change of substrates. All variables need to be cleared upon run complete with the exception of the process start event, for which clearing is not applicable.

Table 1

Data Element	Description	Satisfied by
Load position	The position occupied by a specific wafer within a specific process chamber, i.e. position on the wheel in an ion implanter, location in a furnace tube, etc.	SEMI E90 SubstrateID, SubstLocID, SubstState, and SubstProcState
Load Orientation	Orientation of the notch or other identifying mark on the wafer relative to a fixed point on the wafer holding fixture within the tool.	n/a
Lot# / Material ID	For each load position in the process chamber the tool should provide this data (sourced from MES during run initiation).	SEMI E87 ContentMap, SEMI E40 PRMtrlNameList, SEMI E90 SubstrateID
Slot number / Carrier Position	The position occupied by a specific wafer within the carrier, i.e., Slot x.	SEMI E90 SubstrateID, SubstLocID, SubstState, and SubstProcState. SEMI E94 Control Job, SEMI E87 Carrier Management
Wafer ID	A character string representing the bar code or serial number etched on the wafer (sourced from MES during run initiation).	SEMI E87 ContentMap, SEMI E90 SubstrateID
Process Recipe	The current running recipe in a particular process station.	SEMI E40 RecipeID
Process Sequence	The parent sequence of the current running recipe--applies to cluster tools only.	n/a
Run History	What process recipes has this wafer already "seen". This is to provide 1) setup information for in situ sensors and metrology 2) Allow for fine tuning of FDC models.	SEMI E90 SubstHistory
Process start	Prior to process start, at least by a second or two, the tool needs to provide the process recipe, process sequence, load position, load orientation and lot/slot information to the SECS or EE data interface.	SEMI E40 PRJobState

Upon start of each wafer load process, a “tool processing” event must be supplied. This allows FDC components to activate, configure themselves and any remote sensors, and initiate data collection with a tool. This may be satisfied by the module and/or equipment state of “BUSY” proposed in SEMI Doc. 3296, Equipment Performance Tracking.

The following items are required during a process run:

- Recipe step—An indication of the current recipe step. This status must become zero or null upon run complete.
- Process Parameters—According to the FDC application requirements and the performance requirements of section 3.0.

These are conventionally sent using (S6F1) within the SECS-II specification.

The Process Stop event must be supplied immediately after run complete. At the conclusion of the process the tool must provide notification that the process is complete and the wafer(s) is leaving the chamber. This is satisfied by SEMI E94 and SEMI E40 PRJobState.

The “All Wafers Complete” event must be supplied when all wafers have completed processing. This allows FDC components to shut down, save files, initiate archive actions, etc. This is satisfied by SEMI E87 CarrierAccessStatus.

3.7 Specific Context Requirements for Run to Run Control

Run-to-run control is limited to control that may be applied before beginning a run in response to either the previous run’s results (feedback) or predictive data about the incoming material (feedforward).

The following is required from the equipment:

- Control Recipe Name—Name of control recipe being used. May have modifiers that could indicate specifics of algorithm and control scenario.
- Process Program Name—Used to sync control recipe to process program recipe.
- Process Program Control Variable List—List of variables that can be individually actuated by R2R control.
- Process Program Control Variable Value—(Name, Value) pair used by R2R control system to actuate a process program variable value. Existing SECS messages support this.
- Overwrite Capability—Variable that indicates the capability of the operator to overwrite the control recipe at the tool.
- Overwrite State—Used to indicate that a control recipe has/has not been overwritten at the tool. Indication of overwrite should include values of overwritten variables, passed as “Process Program Control Variable Value”.

The following is required from related metrology system(s):

- Metrology Recipe Name—Used to sync control recipe to metrology recipe. Not necessary in all cases, e.g., if tool provides necessary context linking to associate metrology data with tool recipe.

- Metrology Type—Indication of role of metrology system as in “Pre” or “Post” or “Both”. Not needed if static configuration or if one or other is not available.
- Metrology Variable List—List of variables that can be supplied by the metrology system. This may be a structure that might include frequency of reporting, groupings (e.g., value and goodness of fit), etc.
- Metrology Variable Value—(Name, Value) pair used by R2R control system to actuate a process program variable value. Existing SECS messages support this, however data should have a number of modifiers including quality metrics such as goodness of fit and timestamp, and traceability metrics such as lot/wafer.
- Wafer ID, pre or post, metrology recipe, associated tool recipe, etc. The requirement of these modifiers depends on operational environment.

The following is required from either the MES or a central maintenance system:

- PM Event—Type of PM event that has occurred and possibly other modifiers such as timestamp, run number, etc. This parameter is used by the R2R controller to reset as necessary after a PM event.
- Lot/Wafer ID, etc.—Context information passed to controller with request for control recipe (as necessary), and with report of Metrology data so that controller can link pre/post metrology, control recipe and advice, and process and metrology recipe. Not needed in some straightforward R2R implementations with limited product sets.

3.8 Dynamic Data Context

In several cases, data context associations may be made post mortem outside the equipment from multiple sources. However, there is a set of dynamic context known by the tool control system. In these cases, the tool is the best agent to make these associations accurately. The most fundamental context data item is the data timestamp. Other types of context of interest to APC applications are tool resources, material, and operations.

3.8.1 Timestamp

The most fundamental context data item is the data timestamp. All data must have an associated timestamp. For event data, the timestamp must be accurate to within plus or minus 0.05 seconds of the actual event time. For an equipment system with more than one internal clock, these clocks must be synchronized to provide this accuracy. The equipment must support synchronization of its clocks or a master clock to an outside time source.

3.8.2 Tool Resources

Data must be associated with a tool identity that is unique across the fab. Where a session or connection of data is presumed to apply to one and only one tool at a time, the tool may be identified at the initiation of that data session. Hence, it does not need to be tagged on every message.

For data that applies to a particular process station, that process station must be identified in a manner that is unique within the tool. Location ID is a sufficiently specific identifier. Where ModuleID is also sufficiently unique, Module ID is acceptable. Data that applies to LocationIDs or ModuleIDs that are not process stations should also be correctly identified, but this is not a requirement of APC.

3.8.3 Material Resources

Where a substrate or substrates is present, the identity of those substrates must be associated with the data. This information must be sufficient for an outside APC application to uniquely match data for the same substrates from non-tool data sources, such as the MES, metrology, or central maintenance system. Below the level of substrate, certain APC applications may also require specifics for field, die, die scan, or wafer coordinates.

An absolute identity of the material, e.g., an individual wafer barcode, customer lot ID, etc., is highly recommended to a relative reference, e.g., slot ID, carrier ID. A set of multiple items may be expressed as a list data item.

3.8.4 Operations

Sufficient information must be available to uniquely associate recipe runs and recipe steps with parametric data. The identity of each recipe run must be established to a degree that not only supports APC applications but also tracing of the source instructions for that recipe. In some cases, process program ID may be sufficient.

In other cases, additional information, such as file path name, recipe creation data, etc., must be specified or at least referenced in order to uniquely identify the recipe. The recipe may also be modified by the values of variables set at run time rather than at the creation time of the recipe. Any adjustable recipe variable must be available for reporting at the start of the recipe run.

The recipe step durations are often variable with time, so it is necessary for APC timing requirements that the boundary events of each recipe step are available for reporting.

3.9 Static Data Context

APC data is not required to be self-describing. There are instances of data that could (and should) be tagged outside the tool. For example, the units, resolution, and accuracy of a parameter are generally static, and hence should not be transferred on every message. The static attributes of each variable should be represented in documentation or a discoverable interface and should not be attached to every message.

3.9.1 Parameter and Event Context

For each parametric value available for reporting, there are several static properties underlying that value that may affect the outcome of an APC application.

The units of measure must be documented for all parameters. Even if a parameter is truly unitless, it must still be specifically documented as unitless. Compound units of measure, e.g., cubic centimeters per second, must be documented in a manner that supports automated dimensional analysis.

The resolution and accuracy of each parameter must be documented. If the value is passed in any special format, this format must also be documented.

For each event, a property must be documented specifying whether that event is generated by “control” or “response”. A control event is generated by the equipment control system to motivate an action and is independent of the actual execution of the action. A response event is detected or measured by the equipment control system through any of its input systems, e.g., sensors, GUI, communication ports, etc. If there is a distinction in timestamp accuracy between control events and response events in general, this distinction should also be documented or reported individually.

3.9.2 Data Refinement

Data refinement is the degree to which data has been altered or processed in order to yield information and dictates how it should be interpreted for decision making and/or subsequent analysis. Refinement does not include the capability of data compression. Refinement also does not imply a position along factory information chain; data that comes directly from equipment may exemplify any level of refinement.

1. Raw Data—Direct reading of sensors or other parameters. No aggregation, simplification, calculation, or other conditioning is performed on the data.
2. Conditioned Data—Data that is altered in a manner that supports more meaningful interpretation, e.g., conversion to engineering units, data smoothing. Filtering or sub-sampling of data is a type of conditioning. Running averages or accumulated statistics (e.g., min, max, etc.) are also forms of conditioned data.
3. Derived Data—Data that is derived by equation, formula, or algorithm from other data that does not summarize or identify a feature or features of interest, e.g., converts a data set from time domain into frequency domain.
4. Feature Data (Extracted Feature Data)—Data is processed through an analysis engine that identifies or summarizes a feature or features of interest, e.g., statistical excursions, multivariate analysis, shifts, slopes, spikes, etc.

5. Interpreted Data—Data has been input to a decision or analysis engine and a result of that action is created. This result can be simple, e.g., pass/fail or go/no-go, or complex.

3.9.3 Data Purpose

Data purpose is the category that establishes the reason for collecting the data. The following definitions of data purpose are for user-supplier discussion purposes, e.g., the notion of content, coverage, integrity, accuracy, etc. should be discussed separately for each category. Coverage in each category is highly recommended.

3.9.3.1 Physics Data

Data representing quantities that are used to describe the operating principals or "physical models" behind a process or piece of equipment. In all cases of physics data, the model must be documented or referenced. In the case of secondary physics data, the derivation method must be known and tractable. Subtypes are:

- Process Physics Data versus Equipment Physics Data — *Process physics data* relates directly or indirectly to the processing of substrates. *Equipment physics data* relates only to the operation of the equipment and is independent of substrate processing.
- Primary Physics Data versus Secondary Physics Data —Data that provides these quantities directly may be regarded as *primary physics data*. Data that may be used to derive these quantities may be regarded as *secondary physics data*.

3.9.3.2 Environmental Data

APC applications commonly use environmental data to improve their effectiveness. Environmental data describes the nature of consumable and non-consumable inputs to the equipment and incidental conditions in and around the tool. Measurement examples include atmospheric pressure, temperature, humidity, and vibration. The notion of input purity is confined to this data type.

- Resource Input Data—Data that describes the nature of purposeful input resources to the tool, e.g., input power characteristics, input process gas flows, input fluid flows.
- Tool-Internal Environmental Data—Environmental conditions that are internal to spaces within the tool, either in sealed process chambers or in non-sealed spatial partitions. This shall include areas under local environmental control or areas that are sensitive to intrusions from the general fab environment.
- Tool-External Environmental Data—Incidental environmental conditions that are external to spaces within the tool, but close in proximity to the tool as to potentially affect tool performance, e.g., not the temperature on the other side of the fab.

3.9.3.3 Health Data

Health data is used to characterize the fitness of a tool, system, subsystem, etc. Some APC algorithms react to such data and can adjust controllable parameters to improve performance. The main form of health data is the output of all on-tool diagnostics including component tests, subsystem tests, and end-to-end system tests. Health data also includes certain exception events, such as a signal variation in excess of a predetermined amount, a sensor overshoot or under shoot to desired set point, etc. Anything the tool knows about its own maintenance events and status can be regarded as tool health data.

3.9.3.4 Assurance Data

Assurance data has no direct use other than to verify, validate, or otherwise substantiate the performance characteristics of other data, e.g., confidence intervals, precision tolerances, etc. However, it is nearly impossible to assert confidence in equipment data without having available evidence in this category.

3.9.4 Discoverable Interface

Static data attributes are prime candidates of information that can be served through a discoverable interface rather than through normal data transmission. The following capabilities apply to the concept of a discoverable interface, wherein an outside agent may connect to the tool and automatically discover:

- the intended function of the tool, e.g., an etch tool
- process capabilities of the tool including in-situ metrology
- the configuration of the tool, including the number of load ports, process stations, substrate locations, etc.
- data capabilities of the tool including overall bandwidth, data grammar, and customization
- the event and parameters (including sensor values) available for reporting
- the names, values, and units of equipment constants and calibration data
- the current configurations of data reported from the tool

A discoverable interface can be an effective tool for reducing the cycle time to implement APC applications.

4 Data Quality

This section presents some suggestions for achieving data quality for APC in the areas of data availability, data reliability, sensor tracking and self-checking, and bandwidth monitoring.

4.1 Data Availability and Reliability

The user setting up an APC application will always need to determine that sufficient data and capabilities are available, reliable, of sufficient resolution, and accurate. The supplier shall document all of the data that is available for data collection, including limitations in the reporting of that data. IC makers will evaluate potential availability with respect to this documentation. IC makers and suppliers will evaluate actual data availability by *testing* against this documentation.

Event transmission must meet pre-specified reliability requirements, where event reliability includes both of the following notions:

- an event occurs and that event is not reported or not reported timely and accurately
- a report is transmitted for an event that did not occur.

4.2 Sensor Tracking and Self Checking

Many equipment failures involve sensor failures. This is especially true in modern control systems. Sensor health is essential to proper equipment operation. Tracking individual Sensors will enable troubleshooting and diagnostics. All Sensors providing data shall provide a means of tracking their make, model, serial number, and last calibration date. It is highly recommended that all process-critical sensors should provide a means of automatically verifying proper operation and report the results of such verifications.

4.3 Bandwidth Monitoring

Because the data port used for APC is expected to be frequently stressed, it is important to support monitoring of this vital resource. Monitoring of the data port can be used to troubleshoot problems that occur when setting up large data sets for reporting. The utilization of the port should be indicated on some regular basis, such as an hourly report of data volume and message count. It is also important to understand the peak demand placed on the port by sending a message any time a localized burst rate exceeds a pre-defined threshold.

It is also advantageous to issue exception events related to the data collection system in order to alert an APC engineer, IT technician, or other user that the data may be flawed. Examples would include “data buffer overflow”, “network not responding”, and “latency limit exceeded”.

Appendix A: APC "Learning" versus APC "Doing"

Conceptually, there are two types of APC activities: APC "learning" and APC "doing". The primary use case of APC learning is, through post mortem investigation, to identify unknown relationships and to build models that can be implemented in APC-based solutions. APC learning is executed on a post mortem basis and may involve large quantities of data items that may not have any relation to each other. Data mining is especially suited to aid the problem of APC learning.

Alternatively, APC doing uses automated information and decision-making processes based on known relationships and models to improve overall equipment and process performance. Ideally, a much smaller amount of data is required to operate these models, but the time-based performance of that data is much more critical. APC doing is also dependent on exerting influence on the equipment.

Although there is a significant overlap in the domain of APC learning and APC doing, the technical requirements to support each activity are different, as shown in Table 2. This distinction is intended to help define the real requirements of each activity, and to determine where the solutions for supporting each activity may diverge.

Table 2

	APC "Learning"	APC "Doing"
Data flow	• Many parameters • Large sample rates • Unconditional data collection • Response time not critical	• Fewer, specific parameters • Precise sample rates • Conditional data collection • Critical response time
Database	• Massive storage • Long-term history	• Highly relational/ conditional structure • Critical response time • Critical inter-connectivity between multiple data sources
Example Technologies	• Statistics • Data mining • Principal Component Analysis	• SPC • FDC • Run-to-run control algorithms • Feed-forward algorithms
Modeling	"Textbook" Process Physics	Proprietary Process Models
Tool Control	n/a	• Responsive tool shutdown on command • Adjust input parameters • Adjust real-time parameters • Run-to-run information handling

Appendix B: Features of Extracted Data

This describes some properties for referencing data features extracted over time. The use case is to have a reference pointing back to a larger data set in order to review the validity or consequences of extracted feature information.

- Range Feature—Over a range of time, the data exhibits a particular feature, e.g., slope or shape. A compact reference requires the identities of all component data, timestamps for the start time and end time of the feature's range.
- Point Feature—A feature is identified with an individual epoch in time, e.g., statistical excursion, baseline shift, etc. A compact reference requires the identities of all component data, and three timestamps, one to reference the feature itself and two to define a "neighborhood" about the feature that facilitates clearly viewing the effect of the feature. "Neighborhood" criteria may be subjective.

An extracted feature may exemplify one or more of the following types:

- Polynomial Coefficient—Fitting the data to a polynomial model, a coefficient is calculated, e.g., offset, slope, square, cubic, etc.
- Statistical Parameters—Mean, mode, median, sample variance, confidence interval, etc. The underlying statistical model may be either parametric or empirical.
- Transformation or Decomposition Feature—Feature is not related to data in its natural form, but rather to a transformation or decomposition of the data, e.g., principal component analysis, Fourier transformation, etc.
- Change event—The feature is a change in the value of a more basic feature, e.g., a change or shift in slope, a change or shift in mean, etc.