RTSPC: A Software Utility for Real-Time SPC and Tool Data Analysis

Sherry F. Lee, *Student Member, IEEE*, Eric D. Boskin, *Student Member, IEEE*, Hao Cheng Liu, Eddie H. Wen, and Costas J. Spanos, *Member, IEEE*

***Abstract*—Competition in the semiconductor industry is forcing manufacturers to continuously improve the capability of their equipment. The analysis of real-time sensor data from semiconductor manufacturing equipment presents the opportunity to reduce the cost of ownership of the equipment. Previous work by the authors showed that time series filtering in combination with multivariate analysis techniques can be utilized to perform statistical process control, and thereby generate real-time alarms in the case of equipment malfunction. A more robust version of this fault detection algorithm is presented. The algorithm is implemented through RTSPC, a software utility which collects real-time sensor data from the equipment and generates real-time alarms. Examples of alarm generation using RTSPC on a plasma etcher are presented.**

I. Introduction

TO COMPETE in today's semiconductor industry, companies must continuously improve upon their manufacturing skills to maintain high product quality throughout the entire process. The ability to automatically perform early detection of equipment failures in a production line can lead to significant improvements in the overall capability and profitability of the process.

Recently, there has been tremendous growth in the use of Statistical Process Control (SPC) to generate alarms when the variation occurring on the manufacturing line is unusually large. There are, however, limitations to the effectiveness of traditional SPC techniques when applied to modern semiconductor fabrication lines. First, the data ordinarily used for SPC is often collected long after misprocessing has occurred, causing additional scrap to be needlessly produced from the time of the initial malfunction until its detection. Much earlier detection of equipment malfunctions which cause yields loss will improve the capability and up time of critical process equipment. Second, much of the data available directly from equipment has statistical properties which violate the implicit assumptions used in traditional SPC. Therefore, new techniques are required to improve SPC on real-time data from semiconductor manufacturing equipment.

In a previous publication, we have shown that the real-time data available from sensors in modern manufacturing equipment can be used effectively to detect malfunctions within seconds after they occur [1], [2]. The signals of interest, related to the electrical and mechanical signals within the equipment, are automatically collected while the equipment is processing. In this paper we present a method which has improved detection characteristics and is also suitable for equipment diagnosis.

This improved algorithm is based on a decomposition of the signals and a different time series modeling scheme. Furthermore, the new algorithm has been implemented in RTSPC, a software package which includes automated model generation, data filtering, and a novel double T^2 graphical control chart for the display of alarm conditions. RTSPC interfaces with a workcell controller and can serve as a platform for future real-time process control.

In this paper, an overview of the improved algorithms for real-time SPC is given in Section II. RTSPC, the software platform for implementing these algorithms is given in Section III, followed by an example of model generation and fault detection in Section IV.

Manuscript received May 18, 1994; revised August 9, 1994. This work has been supported by SRC/Sematech (93-MP700), The State of California MICRO Program, National Semiconductor, Texas Instruments, Digital Equipment Corporation, Lam Research Corporation, and IBM Corporation.

S. F. Lee, E. D. Boskin, and C. J. Spanos are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA.

H. C. Liu is with Societé Generale Securities Corporation, NY 10020 USA.

E. H. Wen is with Morgan Stanely and Company, NY 10020 USA.

IEEE Log Number 9407618.

II. Real-Time Statistical Process Control

This section begins with an overview of the improved real-time SPC algorithm. Next, the real-time signals are described, followed by a description of the automatic time series model generator. Finally, the use of the Hotelling's T^2 to combine the multivariate signals into the double T^2 chart is discussed.

A. Overview of RTSPC

This section presents the real-time SPC algorithm. Much of the background information about time series has been previously presented in [2], and will not be repeated here.

1) Baseline Modeling: The real-time SPC methodology utilizes time series models to analyze the real-time signals available from manufacturing equipment through the SECS-II (SEMI Equipment Communication Standard-II) interface. The objective is to use these automatically collected signals to establish the baseline behavior of a complex tool, and later detect deviations from this baseline. Before running RTSPC on production wafers, the following steps are taken to model the baseline condition of the process. The real-time signals from 10 to 15 baseline wafers are first decomposed into long- and short-term components. In single wafer processing equipment, these components represent the wafer-to-wafer averages and the within-wafer signal trends, respectively. Each component

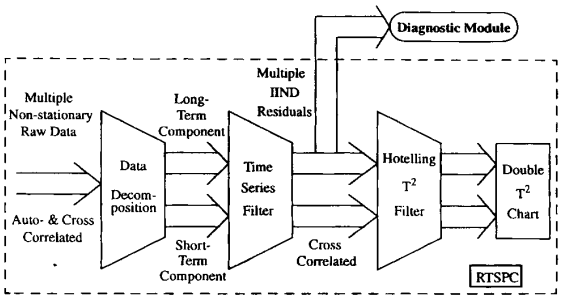

Fig. 1. Real-time SPC data flow.

is then modeled with a time series model. The resulting model forecasts the in-control behavior of the machine.

2) Monitoring the Production Wafers: Once the baseline behavior has been established, production wafers can be run through the machine. As in the training case, the real-time signals from the production wafers are decomposed into the long- and short-term components. Each component is then filtered using the respective baseline time series model. The residuals (the difference between the actual and forecasted baseline values) for each component are then combined using the multivariate Hotelling's T^2 statistic into a single score, which is graphically displayed in the resulting double T^2 control chart.

If no equipment faults are detected, normal operation of the machine continues. When a malfunction is detected, the diagnostic routine is triggered, and an alarm is generated to alert the operator. Diagnosis currently uses the long-term residuals (the difference between the actual real-time signal averages for that wafer and the time series model predictions for the signal averages) as a signature of the specific equipment malfunction [3]. An overview of the real-time SPC data analysis flow is shown in Fig. 1.

B. Real-Time Signals

This section describes the properties of the real-time signals, followed by a discussion of the prefiltering and decomposition performed in the algorithm.

1) Properties of Real-Time Signals: The data collected for fault detection are comprised of various electrical signals such as the radio frequency (RF) impedance and D.C. bias, and mechanical signals such as those signifying the coil and throttle positions. Since the data are collected sequentially at a typical sampling rate of 1 Hz, the signals are correlated in time, demonstrating time series behavior. Time series patterns are observed both within each wafer and across several wafers due to controller adjustments and equipment aging.

Time series signals are highly auto- and cross-correlated. In addition, the correlation structure and the mean value for a given signal may also vary with time, making the series nonstationary. Thus, the data are not identically, independently, normally distributed (IIND), and can not be used directly in a traditional control chart, *e.g.*, a Shewhart chart. Since an underlying assumption of most conventional control charts is that the data is IIND, the first step in the algorithm is to transform the equipment signals to IIND signals. This is achieved by building time series models for each component of each signal. The time series models account for the expected patterns in the data. Once these patterns (whose presence does not indicate a malfunction) are filtered from the signal, SPC can be used to detect deviations in the filtered signals.

The purpose of a time series model is to capture the dependencies among sequential readings of the same process variable. Dependencies within readings collected over time can be described by univariate time series models such as ARIMA (p, d, q) models, where p is the auto-regressive order, d is the integration order, and q is the moving average order. The form of the equation for a nonstationary time series x_t with autoregressive parameters ϕ_k and moving average parameters θ_k is [4]

$$w_t = -\sum_{k=1}^{p} \phi_k w_{t-k} + \sum_{k=0}^{q} \theta_k a_{t-k} \tag{1}$$

where $\theta_0 = 1$, the error $a_t \sim N(0, \sigma^2)$, and w_t are the differenced data

$$w_t = \nabla^d x_t \tag{2}$$

where

$$\nabla^d \text{ is the } d\text{th order of differencing operator} \tag{3}$$

and

$$\begin{aligned} \nabla^1 x_t &= x_t - x_{t-1}, \nabla^2 x_t = \nabla^1 x_t - \nabla^1 x_{t-1} \\ &= x_t - 2x_{t-1} + x_{t-2}, \cdots. \end{aligned} \tag{4}$$

The assumption behind the univariate analysis is that a significant portion of a parameter's behavior can be explained by using past observations of the parameter. A more thorough explanation of time series models is given in [5]–[8].

ARIMA (p, d, q) models can be derived from the collected data when the process is under statistical control; in this way the models describe the baseline behavior of the process. Once developed, the models are used with current readings to forecast each new value. The difference between the forecasted value and the actual value is the forecasting error, or residual. When the equipment is in statistical control, the residuals are by definition IIND variables. As shown in [2], the residuals reflect the equipment state and can be combined in a multivariate control chart to generate alarms.

2) Pre-Filtering of Real-Time Data: RTSPC performs analysis on the main etch step for each wafer. The signals collected during the main etch step are concatenated and filtered as described below. If necessary, the algorithm can be extended to include more than one etch step. The algorithm can also be extended to monitor the length of the etch step (as an additional "long-term" parameter of the wafer) and produce an alarm if the process step takes too long; for example, if an etch step did not endpoint correctly.

Characteristics of the real-time signals caused by transient effects during processing must be accounted for before statistical analysis. At the beginning of processing for each wafer, for example when RF power is applied, a small transient occurs

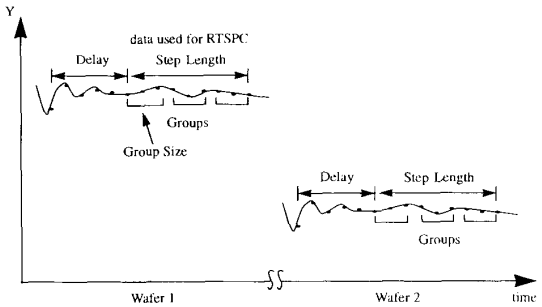

Fig. 2. Real-time signal filtering parameters.

while power is stabilizing. For SPC purposes, the analysis is delayed by a few seconds until the power has stabilized. The delay time is based on the stabilization time for a normally processed wafer. If the RF power, or any other monitored signal does not stabilize in the specified time, an alarm will be generated. To simplify the time series model building process, the same number of data points, or step length, is used for each wafer. Finally, to compensate for the noise in the signals, local averaging is performed within each wafer. The number of samples used in local averaging, or group size, is used to adjust the sensitivity of alarm generation. The delay, step length, and group size are illustrated in Fig. 2.

3) Signal Decomposition of Real-Time Data: As mentioned in the Section *Properties of Real-Time Signals*, time-series models of baseline equipment sensor data are used to filter the nonstationary and autocorrelated patterns in the data. The algorithm presented in [2] builds one seasonal ARIMA (SARIMA) model for each sensor variable.[1] A major disadvantage of this algorithm is that false alarms often occur at the start of a wafer. While these false alarms can be anticipated and ignored, the new algorithm solves this problem more formally.

First, SARIMA models are not appropriate to model the real-time data, because as described in the Section *Pre-Filtering of Real-Time Data*, the pre-filtered wafer signals from the main etch step are concatenated together. This concatenation means the data do not form a natural continuous stream. One assumption behind the SARIMA model is that the variance and the mean of the filtered residuals is the same regardless of the season. Since the discontinuity violates this assumption, the idea of seasons is eliminated in the new algorithm.

The most significant change in the algorithm is the decomposition of the real-time signals from each sensor into long-term and short-term components before modeling. This decomposition is necessary because each component describes a different behavior of the process. An example of signal decomposition of the impedance signal for several wafers is shown in Fig. 3. The long-term component, comprised of the average value of the signal for each wafer, models the overall trend across a number of wafers. On the other hand, the smaller deviations within each wafer create the short-term component, which captures the short-term patterns during the processing of each wafer. Most importantly, the variation of the long-term component is much larger than that of the short-term component, illustrating the point that the short-term components are more sensitive to faster equipment fluctuations, while the long-term components reflect longer duration changes in overall equipment state. This decomposition of the signals into components with drastically different variances is the primary reason the false alarm rate has been decreased. After the decomposition, both components are demeaned to simplify later calculations.

Notice that the short-term component for each wafer in Fig. 3 roughly follows a downward trend. This trend, modeled by the integrative part of the ARIMA model, is captured for each wafer so that deviations from this trend will be detected. Deviations in each of the components reflect different changes in equipment state. For example, a shift in RF power that lasts the duration of the wafer each will be seen as a shift in the long-term signal. A short spike in RF power, however, will be exhibited in the short-term signals. As another example, a dirty film on the wafer is seen by the short-term signals but not by the long-term signal. Because the decomposition allows us to model two different types of faults, the resulting algorithm is more robust than the original method, gives significantly fewer false alarms, and generates residuals that are much more suited for diagnosis.

C. Automatic Time Series Model Generation

Automatic time series model generation, a key module in the new RTSPC software package, makes the difficult and usually tedious process of generating models transparent to the user. In this section, the automatic time series model generation algorithm is described.

1) Automatic Model Generation: Time series models are typically generated interactively using sophisticated statistical analysis tools. The process can be time consuming and tedious, requiring specialized skills to choose statistically significant models. Because models are built for each component of each signal for every recipe, the model generation process can be labor intensive and time consuming. The automatic model generator developed makes this difficult modeling step transparent to the user of RTSPC, thus making the software practical for the factory floor. It is also very fast, typically taking less than one minute on a modern workstation to generate a complete set of models.

Automatic model generation is achieved through several stages. First, if the series is nonstationary, the data is differenced until stationarity is achieved. Next, both the order and values for the autoregressive coefficients are found. Finally, the moving average order is calculated, and an optimizer is used to solve for the moving average coefficients. These steps are outlined below.

To determine whether the series is stationary and requires differencing, the autocorrelation function is calculated. If the first few absolute student-t values of the autocorrelation function drop slowly, differencing is required [5]. This procedure is repeated until the series becomes stationary.

[1] Time series exhibiting periodic variation are said to have seasons, and can be modeled with SARIMA models.

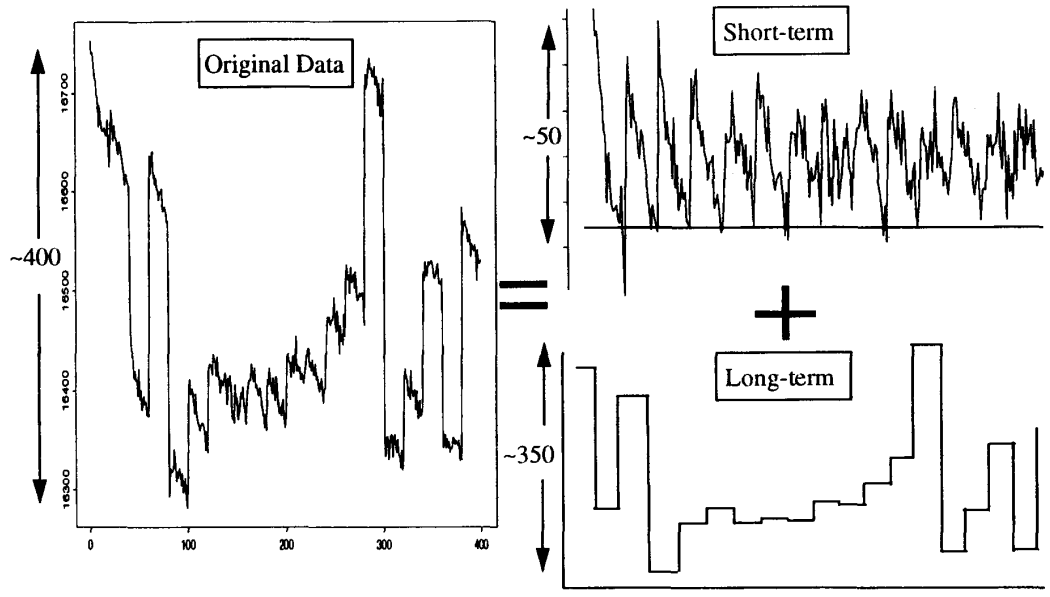

Fig. 3. Real-time signal decomposition.

Next, the modified Yule–Walker equations are used to determine both the order and the value of the autoregressive coefficients. As adapted from [9], the modified Yule–Walker equations are derived starting from the ARMA model. In the following expressions it is assumed that w_t is a real, causal stationary time series that can be modeled with the form shown in (1) and the autocorrelation of the white noise error a_t is defined as $R_{aa}(k) = \sigma^2\delta_k$.

Multiplying both sides of (1) by w_{t-l} and taking the expectation, we obtain

$$R_{ww}(l) = -\sum_{k=1}^{p} \phi_k R_{ww}(l-k) + \sum_{k=0}^{q} \theta_k R_{aw}(l-k) \quad (5)$$

where

$$R_{aw}(m) \equiv E(a_t w_{t-m}). \quad (6)$$

Since the error is white noise and is uncorrelated with future values of w_t, it follows that $R_{aw}(m) = 0$ for $m > 0$. Therefore, in (5), we set $R_{aw}(l-k) = 0$ for $l > q$ and obtain an explicit set of equations used directly to solve for both the autoregressive order and coefficients. To accomplish this an initial value is chosen for q. The only requirement at this stage is to choose a value equal to or greater than the actual number of moving average terms in the final model. Since the actual value of q is not known at this stage, it is safe to choose a fairly large initial value for q at this point

$$R_{ww}(l) = \begin{cases} -\sum_{k=1}^{p} \phi_k R_{ww}(l-k) + \sum_{k=0}^{q} \theta_k R_{aw}(l-k), \\ \qquad \text{for } l = 0, 1, \cdots q \\ -\sum_{k=1}^{p} \phi_k R_{ww}(l-k), \\ \qquad \text{for } l = q+1, q+2, \cdots. \end{cases} \quad (7)$$

For the case of $l = q+1, q+2, \cdots, q+p$, (7) can be rewritten in matrix form. This system is known as the modified Yule–Walker equations [see (8), shown at the bottom of the page]. As shown in (8), the modified Yule–Walker equations relate the ARIMA parameters to the autocorrelation function of the series, regardless of the MA behavior of the process, as long as q in (8) is chosen to be equal or larger than the actual q of the process. When the dimension of the matrix exceeds the number of autoregressive coefficients the matrix becomes singular and its determinant is zero. Therefore, one method often used to choose the autoregressive order of the model is to assume a large number for q so that (8) holds. Then starting with a small number for p, p is increased until the determinant equals zero, or is sufficiently small. Once the autoregressive order has been found, the coefficients can be calculated by solving (8) [9]. This method, however, is not robust in the presence of noise because it is difficult to choose a cut-off point for the determinant, so an alternative method was chosen.

$$\underbrace{\begin{bmatrix} R_{ww}(q) & R_{ww}(q-1) & \cdots & R_{ww}(q-p+1) \\ R_{ww}(q+1) & R_{ww}(q) & \cdots & R_{ww}(q-p+2) \\ \cdots & \cdots & & \cdots \\ R_{ww}(q+p-1) & R_{ww}(q+p-2) & \cdots & R_{ww}(q) \end{bmatrix}}_{\mathbf{R_{ww}}} \times \begin{bmatrix} \phi_1 \\ \phi_2 \\ \cdots \\ \phi_p \end{bmatrix} = \begin{bmatrix} R_{ww}(q+1) \\ R_{ww}(q+2) \\ \cdots \\ R_{ww}(q+p) \end{bmatrix} \quad (8)$$

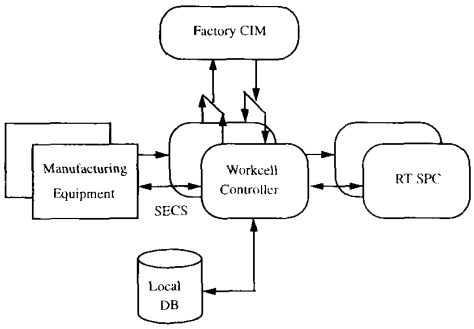

Fig. 4. RTSPC software system environment.

Equation (7) is used directly to obtain both the autoregressive order the coefficients. For a large initial guess for q (which must be greater than or equal to the actual value of q), a linear regression model using (7) is fitted to the time series data. To determine the order of the model, the significance of the coefficients is calculated using the student-t test. The least significant coefficients are eliminated one at a time until all of the coefficients are statistically significant. This method has been found to be both robust and computationally efficient.

Once the autoregressive order and coefficients are determined, the moving average order can easily be calculated from (7) using the proper autoregressive coefficients. This entails finding the largest integer k for which $R_{ww}(l) + \sum_{k=1}^{p} \phi_k R_{ww}(l-k) \neq 0$. The moving average coefficients are then solved using a nonlinear optimizer. The algorithm used for optimization is the Han-Powell variable metric algorithm, which is fairly efficient for a small number of parameters (under 10) and can easily handle both equality as well as inequality constraints.

2) Applying the Short-Term and Long-Term Time Series Models: Usually, at least 10 to 15 baseline wafers are required for accurate baseline models. For each of the short-term components, the autocorrelation is calculated between adjacent points in each wafer, and then averaged across the wafers. The autocorrelations between adjacent wafers of the short-term components are ignored, which differs from the previously published algorithm [2]. The average autocorrelations across each wafer are then used in the modified Yule–Walker equations to build the models. The time series models of the long-term components are built from the average signal value sampled during the time it takes to process one wafer. The resulting time series model generation follows the steps outlined in the previous section.

If a new point is 3-sigma from the baseline forecasted point during the monitoring of production runs, an alarm is generated and the algorithm replaces the "bad" or faulty point with the forecasted point. Thus, consecutive faulty points are detected and the models retain their baseline behavior. This method is used for both the long- and short-term components.

D. Multivariate Analysis: Double T^2 Chart

In production, new observations are compared to the baseline time series model forecasts, creating IIND residuals that can be used in control charts. One set of residuals is generated for each of the long- and short-term components for every monitored real-time signal. Because some of the signals are cross-correlated, using individual SPC charts will show an exaggerated false alarm rate [10]. Instead, the residuals for each component are combined into a multivariate statistical score using Hotelling's T^2 statistic, which takes into account the correlation among the variables used in SPC. More detail on Hotelling's T^2 statistic in the context of this application can be found in [2].

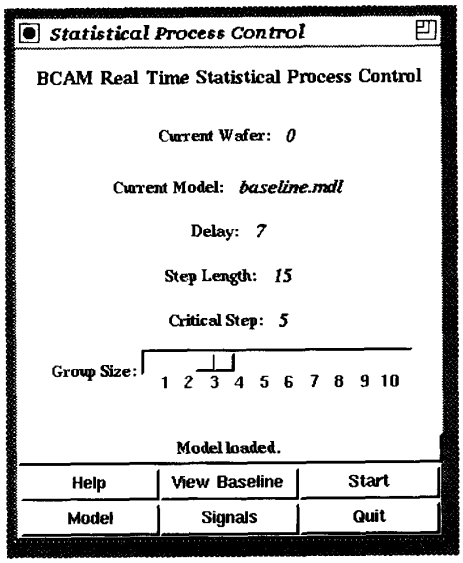

Fig. 5. RTSPC main window.

The resulting Hotelling's T^2 scores for each component are plotted in a one-sided SPC chart. Data points corresponding to run-time faults have residuals which cause the Hotelling's T^2 statistic to be significantly different from zero. One set of scores, obtained from the short-term components, detects faults during the process time of each of the wafers, while the second set of scores, obtained from the long-term components, detects faults by looking at violations in trends across several wafers.

III. RTSPC: The Software Utility

RTSPC is a software application which interfaces with a workcell controller to perform statistical analysis on the real-time sensor data collected from the equipment. An overview of the RTSPC software system environment is shown in Fig. 4. RTSPC communicates with the workcell controller, which collects the real-time signals from the equipment over the SECS communication port. In the current implementation, the controller accumulates the real-time data from the processing of each wafer and then sends a file to RTSPC for analysis. This analysis is typically completed a few seconds after the wafer leaves the chamber.[2]

RTSPC is written for UNIX workstations in a combination of the C and Tcl/Tk programming languages. Tcl/Tk is used for the graphical interface, while C is used for the data analysis.

[2] In the current version of the RTSPC software, a true real-time implementation is only inhibited by data collection logistics.

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.