SNAPSHOT: A PLOT SHOWING PROGRESS THROUGH A DEVICE DEVELOPMENT LABORATORY

Diane Lambert, James M. Landwehr, and Ming-Jen Shyu
AT&T Bell Laboratories
Murray Hill, NJ

December 1, 1992

Codes, processes, and the mix of products at a Device Development Laboratory (DDL) change constantly, and lots may take months to progress through the DDL. Nonetheless, managers need to assess the health of the DDL during short periods and to monitor its health over time, and engineers need to predict completion time and track the progress of their lots, even when there are no previous lots of the same code. This paper presents a new plot of step completion, called a *snapshot*, that addresses both these needs. A snapshot combines information on run times and queue times from existing data bases to show step completion for a code of interest even if no lots of that code were completed during the time period monitored.

EXECUTIVE SUMMARY

Problem

A Device Development Laboratory (DDL) is a facility that develops new codes and manufacturing processes. Because its objective is to learn about codes and processes rather than to produce wafers, lots do not move smoothly through the DDL from start to finish. Some lots are split, some are held for engineering tests, and others are removed after partial processing. Even lots for codes that are nearly ready to be transferred to production may take weeks to finish. Consequently, the mix of codes at the DDL changes constantly as processes are modified, abandoned, or moved to a factory and as new manufacturing processes and codes are introduced.

Despite all the changes and disruptions, engineers need to predict the time required to finish a lot of a particular code and to track the actual progress of current lots relative to predicted progress, even if the code has never been produced before. And despite the lack of standard products and lack of stable conditions, managers need to monitor the DDL's performance over time.

Solution

A code is processed in a sequence of detailed *steps*, such as bake for a specified time at a specified temperature. Ignoring engineering holds, which are delays requested by the design engineer and are not under the control of the DDL, the total time needed to process a lot of a particular code is the sum of the step times, where each step time is the sum of the queue time spent waiting to begin the step and the run time spent in processing the step. Our goal is to characterize the total time needed to process a code of interest through a particular step.

Our approach is to extract queue times and run times for each step needed by the code from existing databases. Because each step is specified in detail, run times are the same, apart from random lot-to-lot variability, for all lots of *all* codes that pass through the step. Queue times for a particular step depend on the priority of the lot, but are the same for all lots of the same priority, apart from random lot-to-lot variability. Thus, there can be data relevant to the code of interest, even if no lots of that code were produced from start to finish during the time period being monitored.

From the data on all lots that pass through a given step, we estimate the mean and standard deviation of queue time and the mean and standard deviation of run time for that step. Then we use cumulative sums of the estimated step time means and variances to describe the time needed to process lots from the code of interest from the start through each step.

The cumulative times are displayed in a snapshot plot, which represents a particular code of interest and a specific time period of DDL operation. It plots the cumulative mean time to complete processing through a step on the vertical axis against step number on the horizontal axis. Thus, for example, the mean time to complete all processing is shown above the last step number in the code. The snapshot also shows two curves that represent optimistic and pessimistic time estimates. The lower curve shows the estimated time required for completion through each step by the fastest 5% of lots of the code of interest, plotted against step number; the upper curve shows the estimated time required by the slowest 5% of lots.

Conclusions

A snapshot of step completion shows the expected progress for a lot of a particular code if the DDL continues to operate as it did during the time period covered by the snapshot. Since the required information on run times and queue times is obtained from all the lots that pass through a step, regardless of their codes, a snapshot can be produced even if no lots of the specified code were completed during the period that the snapshot describes.

Snapshots have two important uses. Snapshots for the same code but different time periods can be used to track the overall operation of the DDL over time. This comparison is valid even if the mix of products in the DDL changes, as long as the steps for the code are not redefined. In addition, a snapshot can be produced for a particular code when a lot enters the DDL, as long as there is information on the run times and queue times for each of its steps. The snapshot will show the mean predicted time and optimistic and pessimistic predictions of the time needed to process the lot; the lot's actual time to complete each step (excluding time on engineering hold) can be added to the plot to monitor its progress.

THE PROCESS

Yield and standard measures of throughput, such as cycle time, are sensible measures of performance for a production line making at most a few different codes. Farrell et al. (1992) discuss measurement and analysis systems to monitor and improve the efficiency of production lines. But overall yield and standard measures of throughput may be inadequate for a DDL that develops new codes and new processes in a rapidly changing environment. For example, one standard throughput measure is the total or average number of steps completed per day in the DDL by all wafers in all lots. The rationale is that steps are easy to count, and the more steps completed, the more productive the facility. In a manufacturing facility with little change from day to day, comparing step counts from day to day makes sense. But when steps differ in complexity and time required and the environment changes rapidly, the number of steps completed is of limited value for tracking the DDL as a whole or for monitoring the progress of specific lots.

Because the contents of individual steps in a DDL are more stable than codes and sequences of steps, we believe that performance measures for a DDL should consider the information from each step separately rather than information obtained by considering all steps together, without regard to their differences. While information from individual steps, such as step yield and step run times, is important for monitoring DDL activity, information from the separate steps also needs to be combined to assess the DDL's performance as a whole. This last aspect of performance is the one addressed by the snapshot plot in this paper. Namely, the snapshot is a way to extract information from the separate steps and combine it to understand DDL output as a whole without following lots of one particular code.

The key idea is that different codes have different, lengthy sequences of steps, but often steps are shared by several codes. Moreover, all lots of the same priority and size that pass through a particular step, regardless of their code, should require the same queue time and run time, except for the random lot-to-lot variability that pertains to the step. Thus, instead of measuring throughput by tracking lots of a particular code from start to finish of processing, we measure completion time for individual steps and then sum completion times for individual steps.

THE DATA

For each lot entering a step, the number of wafers at the start of the step, the time of entry into the queue, the start of processing for the step, and the end of processing for the step are routinely reported to a database. From these times, a queue time and run time are calculated for the lot, taking into account holiday and weekend periods when the DDL does not operate. The number of wafers that can be processed simultaneously in each step, called the *batch size*, is also reported.

Engineers and data base managers at a DDL chose a code, and for each step in the code provided the run times and queue times for all lots that used the step in a six week period. There were 148 steps in the code, but some steps, such as cleans, were used more than once. Five steps had been used by only a few lots during the six week period; the engineers replaced these steps by other similar ones that had more data. In the end, we had data for a code that had a total of 148 steps, some of which were repeated. There were 120 different steps in all. Lot priorities were not provided, but most lots observed were believed to have the highest priority. Queue times depend on the lot's priority, but run times do not.

The observed run time for each lot was converted to a unit run time by dividing the lot run time by the number of batches in the lot. For example, if 6 wafers can be processed simultaneously in a step and there are 40 wafers in the lot, then the batch size is 6 and the unit run time is the lot run time divided by the number of batches of size no more than 6, which is 7 batches.

Queue times and lot run times were calculated from data manually entered into the database, and so were sometimes unreliable. For example, an operator should time stamp the lot manually when a step begins to run, but sometimes does not. If the time that processing ends must be entered before the lot can move to the next step, the technician may enter the start and end of processing simultaneously, giving an incorrect run time of zero. Not all zeros are erroneous, however. When only a random sample of the lots is inspected, many of the observations for an inspection step are zero.

There were also unreasonably large unit run times for some steps. For example, one cleaning step had a unit run time of more than 8 days, when its median unit run time was 40 minutes. In practice, reasonable minimum and maximum unit run times should be provided by process engineers. In this test case, we identified aberrant run times informally ourselves, rejecting only values that were extremely far from the bulk of the data.

ANALYSIS AND RESULTS

The snapshot plot requires estimates of the mean and variance of lot run time and queue time for each step of a code. In this section we first show how we estimated the means and variances for each step of a code using the data provided by the DDL and then we show how to construct a snapshot.

Estimated Means and Variance

First consider unit run time. (The discussion for queue times is similar.) The obvious estimates of the mean and variance of unit run time for a step during a particular period are the sample mean $\bar{X}$ and the sample variance S^2 of the observed unit run times $X_1, \ldots, X_n$ during the period. The sample mean and variance are attractive because they are widely understood, unbiased, and the best possible estimators if the data are normally distributed.

But if the data are not normally distributed, or if the data are normal but include frequent outliers, then there are better estimators of the true mean and variance than $\bar{X}$ and S^2. Sometimes it is obvious that a value is an outlier and should be ignored before making any calculations. For example, if only one unit run time for a step exceeds 40 minutes, and that unit run time is 8 days, then we would ignore the 8 days. But if a unit run time is 35 minutes when all other run times are between 20 and 30 minutes, it is not obvious whether this value is an error that should be ignored. While the data might follow a normal distribution with this value an outlier, the data might instead follow some non-normal distribution, in which case better estimators could be used. Instead of using $\bar{X}$ and S^2 as estimators along with some rules for eliminating possible outliers, we prefer to estimate the mean and variance by using the parameters of a distribution that fits the data. Choosing an appropriate distribution may require the help of a statistician, but once it is chosen it is straightforward to update the estimates of its mean and variance.

In our study, we found three standard distributions to be adequate for modeling all the queue times and unit run times: the normal, lognormal, and Weibull (Johnson and Kotz (1970) discuss these distributions). As Figure 1 shows, normal observations are symmetric around the mean, but lognormal and Weibull observations are not symmetric and large observations are likely. The logs of lognormal data are normal while the logs of Weibull data may be asymmetric and have a right tail that is longer or shorter than that of a normal. Some lognormal and Weibull densities are similar, however.

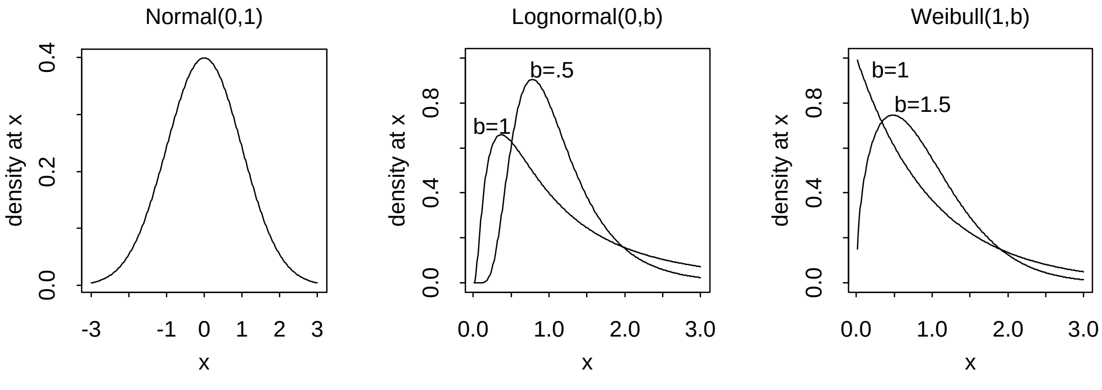

Figure 1: Three Families of Densities Used to Model Run Times for the DDL Data

A lognormal(a, b) distribution has mean $A\sqrt{B}$ and variance $A^2B(B-1)$, where $A = e^a$ and $B = e^{b^2}$. A Weibull(a, b) distribution has mean $a\Gamma(1+b^{-1})$ and variance $a^2\Gamma(1+2b^{-1}) - a^2\Gamma(1+b^{-1})^2$, where Γ is the (complete) gamma function, which is widely available. Thus, to estimate the mean and variance of unit run time, we need to estimate a and b.

For the normal(a,b), the parameter estimates are $\hat{a} = \bar{X}$ and $\hat{b} = S$. For the lognormal(a,b),

$$\begin{aligned}\hat{a} &= n^{-1}\sum \log(X_i)\\ \hat{b} &= \left[n^{-1}\sum \left(log(X_i) - \hat{a}\right)^2\right]^{1/2}.\end{aligned}$$

For the Weibull, the maximum likelihood estimators solve

$$\begin{aligned}\hat{a} &= \left(n^{-1}\sum X_i^{\hat{b}}\right)^{1/\hat{b}}\\ \hat{b} &= \left[\left(\sum X_i^{\hat{b}}\log(X_i)\right)\left(\sum X_i^{\hat{b}}\right)^{-1} - n^{-1}\sum \log(X_i)\right]^{-1}.\end{aligned}$$

These equations are easy to solve by iterating for $\hat{b}$ and then calculating $\hat{a}$.

Once (a,b) are estimated for the normal, lognormal and Weibull, we have a fitted normal, fitted lognormal and fitted Weibull distribution for the data. To choose one of these, we plot the ordered data against the $(k+.5)/(n+1)$, $k = (1,\ldots,n)$, quantiles of the fitted distribution. These quantiles act like the expected order statistics for a random sample of size n from the fitted distribution (David 1981). Therefore, if the points in the quantile plot follow a straight line with an intercept at 0 and a slope of 1, the fitted distribution is a good model for the data.

Figure 2 shows the fits for an etch step that was used 20 times during the 6 week period of interest. The solid line in each panel is the reference line with intercept 0 and slope 1. For this step, the lognormal does not fit as well as the normal or Weibull, because it overestimates large observations. That is, the right tail of the lognormal is too long. There is little to distinguish the normal and Weibull fits, so we choose the normal because it is simpler. We also choose the normal when there is little data. For example, some steps were used only 4 times during the 6 week period.

Figure 3 shows the normal, lognormal, and Weibull fits to an apply step that was used 110 times during the 6 week period of interest. Each observation but one was below 35; we omitted a large outlier which was above 100. Here the Weibull provides the best fit. The lognormal seriously overestimates the larger observed run times, and if used it would overestimate the mean and standard deviation of unit run times. The normal fit underestimates the small run times and the large run times, so if used it would underestimate the mean unit run time.

The lognormal and Weibull distributions assume that the data values are all positive and can extend as low as zero (see Figure 1), but for some steps very short run times are clearly unreasonable. For these cases we shift the lognormal and Weibull distributions to have a larger threshold that is a sensible value for the step, rather than zero. For example, Figure 4 shows 46 observations from a bake step (5 observations of 0 and 5 observations less than 30 were omitted), using a threshold of 180 for the lognormal and Weibull.

Out of 120 steps for unit run times, we chose 50 to be Weibull, 50 lognormal, and 20 normal. Of the 100 that are lognormal or Weibull, only 6 needed a nonzero threshold, and those fits are not too sensitive to the actual choice of threshold.

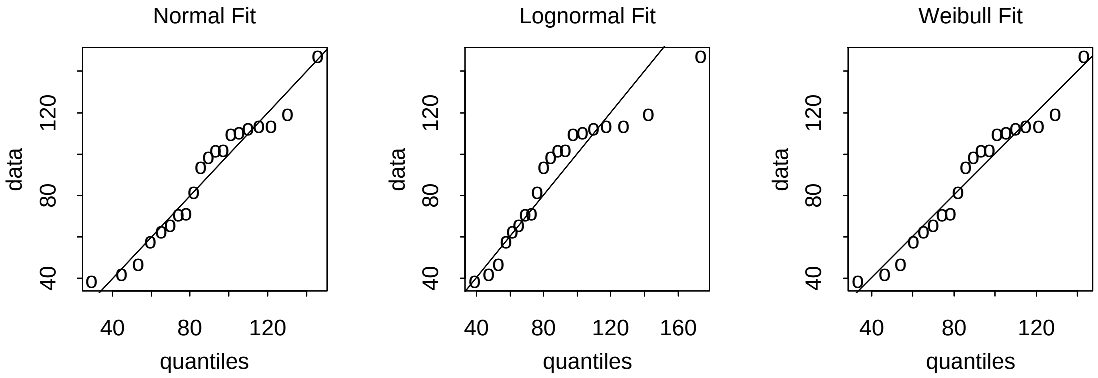

Figure 2: Normal, Lognormal and Weibull fits to the unit run times from an etch step. The right tail of the lognormal is too long, as it overestimates the largest observation. The normal and Weibull fits are similar. For this step, we choose the normal because it is simpler.

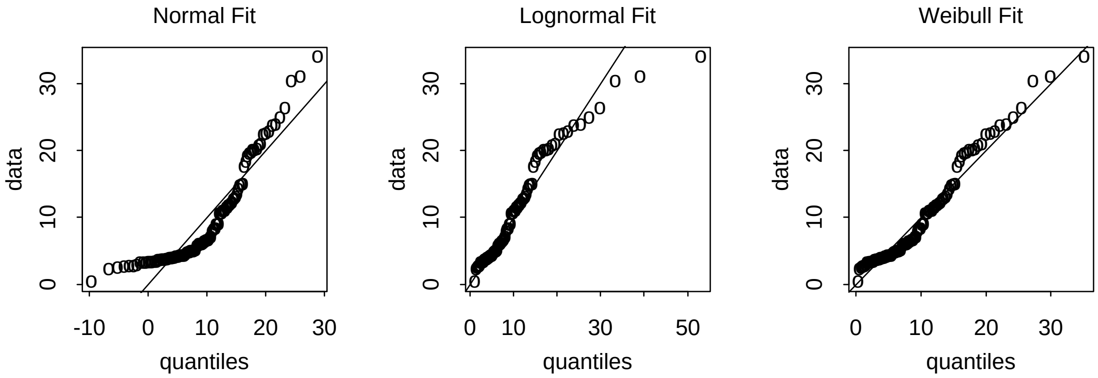

Figure 3: Normal, Lognormal and Weibull fits to the unit run times from an apply step. The normal underestimates both small run times and long run times. The right tail of the lognormal is too long. The Weibull provides the best fit.

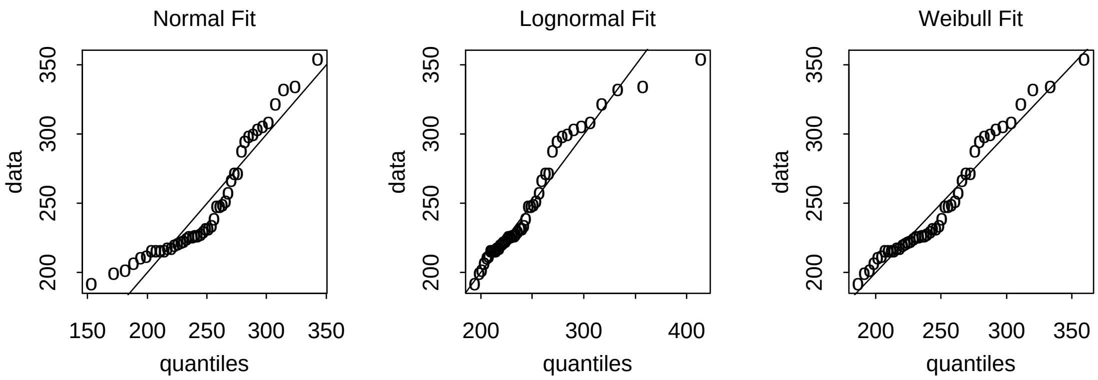

Figure 4: Normal, Lognormal and Weibull fits to the unit run times from a bake step. For the lognormal and Weibull, a threshold of 180 minutes is assumed. The normal distribution does not fit the observations below the mean; the right tail of the lognormal is too long. The Weibull provides the best fit.

Sometimes, a step is applied to only a random sample of lots, so many unit run times are zero. For example, for one inspection step 23 of the 46 unit run times were zero. For such steps, we modeled the distribution of unit run times in two parts. First, with probability p_+ the lot is inspected and the unit run time is positive. The positive run times are assumed to be either normal, lognormal or Weibull. Suppose the true mean and variance of the positive inspection times are μ_+ and σ_+^2. Then the mean wait for a random lot that might or might not be inspected is the mean wait for an inspected lot times the fraction of lots inspected. The variance of a random lot is the variance within inspected lots (the variance is zero for uninspected lots) multiplied by the fraction of inspected lots plus the variability between the mean of the inspected group (μ_+) and the mean of the uninspected group (zero). That is, the mean μ and variance σ^2 of the unit run times for a random lot that might or might not be inspected are given by

$$\begin{aligned}
\mu &= p_+\mu_+ \\
\sigma^2 &= p_+\sigma_+^2 + (1-p_+)(0-\mu)^2 + p_+(\mu_+ - \mu)^2 \\
&= p_+\sigma_+^2 + (1-p_+)p_+^2\mu_+^2 + p_+(1-p_+)^2\mu_+^2 \\
&= p_+\sigma_+^2 + p_+(1-p_+)\mu_+^2.
\end{aligned}$$

(See De Groot (1975) for more discussion of conditional means and variances.)

The probability p_+ of a lot being inspected can be estimated by the fraction of nonzero observations in the data. The mean and variance of unit run time for inspected lots can be estimated by fitting either a normal, lognormal or Weibull distribution to the run times for inspected lots.

Although the discussion so far has been in terms of unit run times, the same ideas apply to queue times (except that queue times apply only to lots, not to batches within lots). Often, there is a large probability that a lot does not have to wait before being processed, so the queue times must be treated as the run times for randomly inspected lots were treated. That is, the lots that do not wait must be treated separately from those that do wait to find a distribution that fits the data well. Since queue times may be longer for lots with lower priority, the distribution of queue time should be estimated from lots with the same priority.

Finally, note that the same methodology applies regardless of whether the queues between run steps are input queues, with lots waiting at a step before being run in the step, or output queues, with lots waiting after finishing a step and moving on to the next step only when the next step is ready to process it.

The Snapshot Plot

A snapshot describes the cumulative mean time needed to process a code through its steps and optimistic and pessimistic estimates of the cumulative time for processing. The estimated mean time to complete a step is the sum of the estimated mean queue time and mean run time for the lot, where mean run time for the lot is the product of the number of batches in the lot and the estimated mean unit run time for the step. The mean of a sum of steps is then the sum of the means for each step, even if the steps have different distributions or if the distributions are statistically dependent (see De Groot (1975)).

The estimated cumulative time for the fastest 5% of lots depends on the distribution of the sum of step times. Here, the exact distribution of the cumulative time to complete processing through a particular step is the sum of Weibull, lognormal and normal random variables. The distribution of such a sum is difficult to find. Therefore, we simplify the problem in three ways. First, we assume that, for any step, the queue time and lot run time are independent. In particular, lots that wait a long time to begin a step of processing cannot be processed faster than lots that are processed immediately. Secondly, we assume that step times are independent, so if a lot finishes one step quickly it is not more or less likely to finish the next step quickly. Process engineers agreed that both these assumptions were reasonable.

Finally, we simplify our task by using the fact that a sum of independent random variables is approximately normal with mean equal to the sum of the means of the random variables and variance equal to the sum of the variances of the random variables (some weak regularity conditions are needed, see De Groot (1975) for a detailed statement). Thus, the total time needed to complete processing through a step is approximately normal. For a small number of steps, the normal approximation may be poor. But after several steps, the normal approximation should be adequate. We use the .05 quantile and .95 quantile of the approximate normal distribution to estimate the completion time for the fastest 5% of lots and the slowest 5% of lots.

Figure 5 shows a snapshot for a standard code representing a specific six week period of

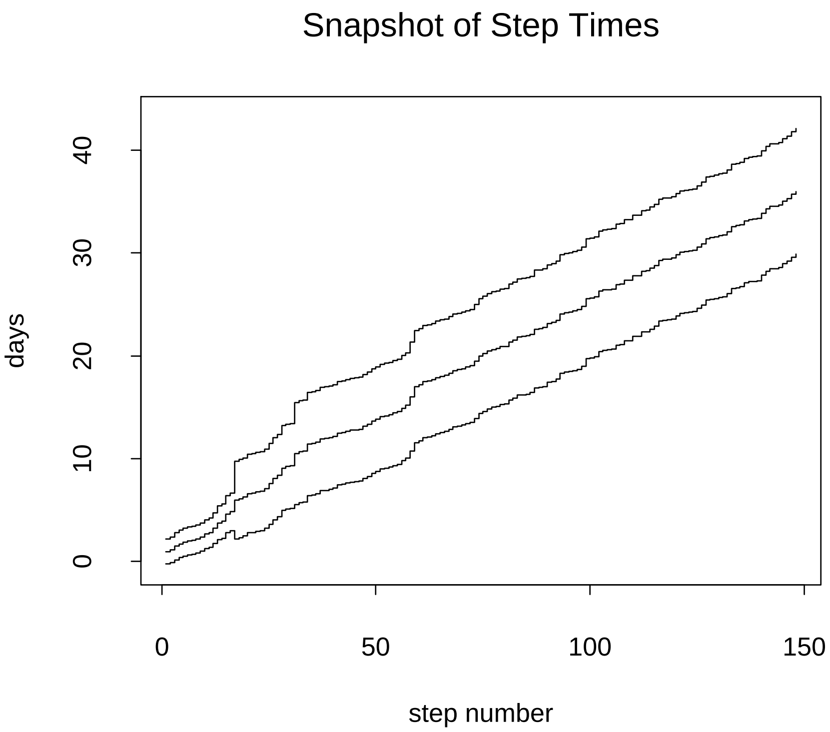

Figure 5: The middle curve shows expected time to complete processing through a step against step number for a particular 148 step code based on data obtained during a particular 6 week period at a DDL. The lot size is assumed to be fixed at 40 wafers throughout processing. The upper curve shows the estimated time to complete the slowest 5% of lots and the lower curve shows the estimated time to complete the fastest 5% of lots.

data, and assuming a constant lot size of 40 wafers throughout processing. The middle curve shows that on average it takes 13.9 days to complete the first 50 steps of the code and 36.0 days to complete all steps of processing. The fastest 5% of lots complete step 50 in 8.8 days and the entire process in 29.9 days, while the slowest 5% of lots complete step 50 in 18.9 days and the entire process in 42.1 days. Note that the optimistic, lower curve is negative at early steps because the normal approximation to the cumulative completion time is poor when there are only a few steps.

To track a DDL's performance over time in terms of this standard code, the snapshot can be re-calculated with data from a different, possibly overlapping, time period. The three curves in the new plot can be compared to those from the earlier plot to determine how the estimated time required to produce this code has changed. To make comparisons reflect changes in distributions of queue and run times rather than time changes resulting from differences in yields, it is reasonable to fix both lot sizes at 40 wafers throughout processing.

To track the progress of a specific lot, the snapshot can be calculated when the lot enters the DDL as long as data is available for queue and run times for each of its steps. If the code requires a few new steps or ones that have not been run recently, process engineers may be able to identify similar steps that have been run recently. The lot's actual time to complete each step can be added to the plot as it progresses through the DDL. Of course, the time should subtract out engineering holds and periods such as holidays and weekends when the DDL is not operating. If a new lot represents a new process with many new steps for which there is not comparable data already available, then a snapshot cannot be calculated to monitor its progress. This is reasonable, because it makes clear that it is impossible to predict processing time time for this lot from past data.

When tracking a specific lot, information on wafer loss can be incorporated as it becomes available. If some wafers do not move to the next step, the snapshot can be re-drawn to reflect the reduced lot size in later steps. It is also possible to provide snapshots for lots used in experiments in which some lots are moved to one step, some to another, and at a later step the split lots are combined. The snapshot can show the expected time to finish the slowest lot or the average expected time to finish the split lots.

CONCLUSION

Snapshot plots are a simple but powerful tool for measuring important aspects of the health of a DDL and the progress of lots through it in an ever changing environment.

ACKNOWLEDGEMENTS

We thank Jay Reightler and Linda Snyder for their help in obtaining and processing the DDL data used in this paper.

REFERENCES

David, H. A. (1981). *Order Statistics*, 2nd. ed. John Wiley & Sons, New York, NY.

Farrell, B. L., Kahan, W., Reightler, J. W. and Woodard, A. C. (1992). Improving service efficiency in manufacturing integrated circuits, *AT&T Technical Journal*, **71**, 37-44.

De Groot, M. H. (1975). *Probability and Statistics*, Addison-Wesley, Menlo Park, CA.

Johnson, N. L. and Kotz, S. (1970). *Continuous Univariate Distributions-1*, Houghton Mifflin Company, Boston, MA.