Knowledge-Based Statistical Process Control

Kenneth R. Anderson,
BBN Laboratories
10 Moulton St.
Cambridge, MA 02138

David E. Coleman, C. Ray Hill,
Andrew P. Jaworski, Patrick L. Love,
Douglas A. Spindler,
Alcoa Laboratories
Alcoa Center, PA 15069

Marwan Simaan
Electrical Engineering Department
Benedum Hall
University of Pittsburgh
Pittsburgh, PA 15261

Abstract

In this paper we discuss a set of software tools developed to support the tasks associated with managing special causes of variation in a manufacturing process. These tasks include the detection of significant changes in process variables, a diagnosis of the causes of those changes, the discovery of new causes, the management of performance data, and the reporting of results. The software tools include automatic recognition of "out-of-control" features in critical process variables, rule-based diagnosis of special causes, a model-based search for symptoms where a diagnosis is not possible, and automated reporting aids. It is hoped that these tools will enhance the efficiency of special cause management.

Introduction

American manufacturing has recently emphasized the use of statistical process control methods to limit process variability and produce higher quality products (Wadsworth, Stephens, &Godfrey 1986). This technology relies on the construction of charts, the observation by a machine operator or engineer that these charts indicate an "out-of-control" condition, a diagnosis of the cause of that condition, and the choice and implementation of a corrective action. We refer to this collection of tasks as "special cause management."

Statistical Process Control (SPC) methods are intended to distinguish a variation in a process signal that is significantly different than the usual variability of the process. The statistical process control model assigns such variation to special causes, events that occur in time that are not part of the normal operation of the process. Such events might include material changes, equipment failures, operator error, or environmental changes. The unexceptional or normal variation is said to be due to common causes of variation.

Distinguishing special cause from common cause variation is a probabilistic decision requiring knowledge of the normal process variability when it is "in control"; that is, when no special cause variation is present. The usual technique is to apply various decision rules to control charts (Shewhart 1926; Shewhart 1931; Maragah & Woodall 1988).

Two practical problems occur which limit the effectiveness of the usual SPC approach. First, in a complex process like aluminum sheet rolling there are several process variables that need to be examined. The need to monitor equipment performance, operator procedures, environmental factors and product properties can produce an overwhelming amount of data. It is difficult for a small group of engineers to regularly examine all the data. Instead, their time tends to be devoted to dealing with current "crises." They can slip into a reactive mode rather than systematically identifying sources of variability and removing them, or reducing their effects. Second, process expertise sufficient to diagnose problems causing out-of-control conditions varies dramatically from plant to plant and amongst engineers. Those skilled engineers who have enough process knowledge to do this work are in great demand. They often find it time consuming and tedious to manually search out causes for problems and record them for correction and historical analysis.

These limitations suggest that a scheme which combines sound statistical principles together with knowledge of the process and software tools for managing a database of past performance might allow a more comprehensive analysis and management of process consistency. A program called PROSAIC (Process Signal Interpreters Assistant) has been developed to explore these possibilities in the manufacturing process which produces aluminum sheet.

The program has two major components. The first is a database and interactive graphics system which gives the user basic tools to examine process data. The second part of the program, which is the subject of this paper, more specifically addresses the process consistency problem.

Figure 1 is a block diagram of the program. A method has been developed which recognizes significant variations in process signals. This method detects signal features that are impulses, mean shifts, and/or trends. Thresholds used in this detection scheme are analogous to limits in the usual control chart schemes. For each of the various types of out-of-control phenomena detected there exists a set of rules which is used to try to diagnose what caused this occurrence. The rules are based on past experience and basic knowledge of the relationships between special causes of variation and the symptoms they produce in signals observed in the process.

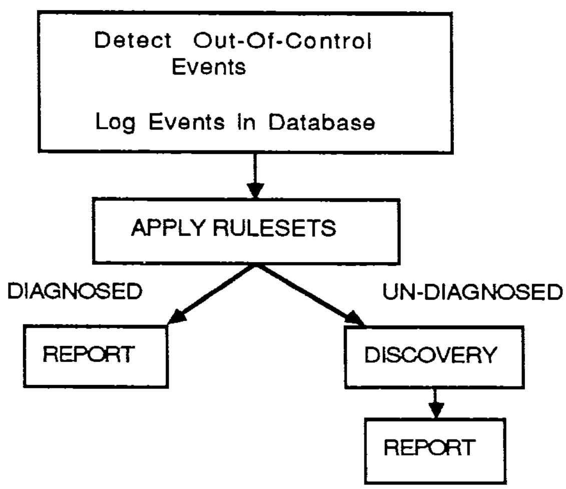

Figure 1. Special Cause Management Block Diagram.

Applying the rules for each set of events results in some events being diagnosed. Information about the diagnosed events and details of their diagnosis are stored in an object-oriented database from which various summary reports can be generated. The remaining un-diagnosed events can be analyzed in two ways. First, an exhaustive application of the ruleset can be applied to inform the user why each of the tested special causes was not asserted as the cause of the problem. This allows the detection of "near misses": cases in which an event nearly meets all the tests of a given special cause apart for some exception. Second, an engineer using the program can call upon an expert's view of the process, stored in a network diagram of influences. A method exists for searching signals in this diagram to identify other signals which were out of control at the same time that the problem occurred. The engineer can use the process model to examine the path along which the associated changes might be influencing the signal of interest.

Tools that operate on the database support the management of special causes. The engineer can examine a Pareto chart of causes for a particular signal and observe the impact of corrective actions. He can also note when new special causes appear that remain undiagnosed and has tools to focus an investigation on what might cause them. Finally, he can automatically generate reports to be sent to other members of the engineering staff so that they are informed about the current state of control of the process.

Detecting Significant Variations In Time Series

Complications arise in monitoring critical process variables that constrain the validity of many control chart schemes. It is usually assumed that successive samples of the process are independently and identically distributed. While a case may be made for this in the manufacture of discrete parts, where charts have been used extensively, in many manufacturing processes there are physical, chemical, and other effects which introduce autocorrelation. Autocorrelation degrades the hypothesis testing prescribed in control charts by changing (sometimes severely) the rates of type 1 errors (false alarms) and type 2 errors (events which did not trigger alarms) (Maragah & Woodall 1988). Furthermore, increasing sophistication of measurement and data acquisition systems has led to higher sampling rates - which increases autocorrelation.

Additional complications arise due to the high cost of a broad class of corrective actions. This means that the process is not often stopped until a regularly scheduled maintenance period. An analysis of out-of-control conditions during the run can and should be used as planning input to the maintenance session. Under this practice, the standard assumption of identically distributed data is violated in cases where out-of-control features which change the process mean occur in the data interval. Unfortunately, conventional control chart detections of out-of-control features after a change in mean are unreliable and often misleading until the chart is "reset." Frequent resetting is impractical.

Manufacturing process signals contain impulsive changes due to process shocks, sudden shifts in the mean, gradual trends, exponential decays to equilibrium and others. Control chart supplementary runs rules attempt to detect some of these variations, but (as stated above) they are compromised by autocorrelation and mean changes (especially when they are superposed). In our work, we have developed a nonlinear signal processing scheme (Love & Simaan 1988) which detects (in the presence of noise) the following three basic features in a process signal: peaks, denoted by P; steps, denoted by S; and ramps, denoted by R.

Peaks are impulses of short duration, steps are shifts in the mean value, and ramps are linear trends in the data. It is assumed that useful information in the process signal can be summarized by this set of three signal features. Note that this is equivalent to modeling the data as a piecewise linear function in time, with added noise from a contaminated distribution.

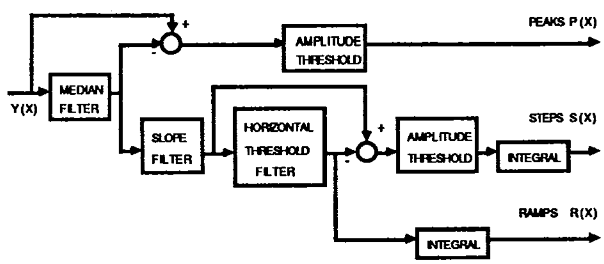

Figure 2. Nonlinear Filtering Scheme to Produce Peaks, Steps, and Ranps from Input Signal y(x).

The essence of this assumption is that these features represent different manifestations of process variations and consequently are sufficient to approximate most process behavior. Should process variation be observed that cannot be described by some combination of these features then the "vocabulary" would have to be extended and other feature detectors would be needed. An example would be a sinusoidal variation.

The automatic interpretation of signals by detecting and analyzing signal features has been reported in several application areas (Stockman, Kanal, & Kyle 1976). This approach to interpretation segments the signal into regions (features) that share common statistics. Syntactic analysis techniques are then applied that treat these features like words in a sentence. More complex signal structures are generated by combining features (Fu 1974).

A block diagram of the signal processing scheme is shown in Figure 2. The input y(x) is the signal to be processed, and the outputs P(x), S(x) and R(x) indicate the three features of interest at sample location x. The basic elements of the scheme are: a median filter, a slope filter, and a horizontal threshold filter.

The median filter is an effective technique for suppressing impulses from a signal (Tukey 1974). It replaces a sample value at location xi by the median of sample values in a window centered at xi. The slope filter replaces the sample value at location xi with the slope of the data in the same window. This slope is determined by a linear least squares fit to the signal values in the window. In both cases the window is moved over the entire length of the data signal. Finally, the horizontal threshold filter is a nonlinear filter which replaces every sample value in the signal by the average value of all samples in a string of numbers provided: (i) the length of the string is larger than a horizontal threshold, and (ii) the average value of the samples in the string is larger than an amplitude threshold.

To set thresholds for this scheme the user is required to pick data intervals that are "in control." These intervals are assumed to be devoid of special cause variation, to collectively contain many points (say, 1000+), and to be indicative of the variability of the in-control process due to common causes. These intervals are passed through the signal processing scheme with all thresholds set to zero. Histograms of the processed data at the point where thresholding is to be applied are generated. These histograms are assumed to represent the likelihood of observing certain values for peak heights, step sizes, slope lengths, etc. in the in-control data; they define values for rare events. Given a user-specified confidence limit, thresholds can be determined by calculating the area under these histograms. This area is the approximate Type 1 error rate. The program can also estimate the Type 2 error associated with these levels for various process shifts.

After features are detected by the nonlinear filters, a list of features is generated and parsed to determine if some subset of features should be combined as a description of a more complex event. The word "parsed" is used to describe the procedure of analyzing features and forming events using certain grammar rules: rules that describe the structure of events of particular types. As an example, define a "thermal," denoted by T, as a sharp step increase followed by a ramp with negative slope (the name "thermal" is chosen because such an event is sometimes associated with thermal disturbances in a process). After the list of features has been scanned, all features that do not participate in more complex events are promoted to the same status as events. The final set of events, formed by the union of parsed events with promoted features becomes the set of detected events for the data interval. Events are represented in the computer as objects in an object-oriented programming system called FLAVORS (Weinreb & Moon 1981). When events are detected they are stored in an object-oriented database.

The operations described above can be performed interactively via a specially designed user interface. These same operations can also be run automatically by a process that starts up at user specified times and generates reports of its activities in both electronic and hardcopy form.

Diagnosing Causes For Out-Of-Control Events.

Rule-based diagnosis systems are commonplace in the literature of applied computer science. (Hayes-Roth, Waterman, & Lenat 1983; Buchanan & Shortlife 1983) Rule-based systems are typically constructed by specifying situation-action or if-then statements. They possess a working memory which keeps track of all facts that have been proven and an inference system that tries to prove new assertions using logical tests on facts already proven. It is assumed that the data to be tested and the knowledge that is used to construct rules are both static and reliable. Such systems work best if the number of possible solutions or diagnoses is small. Rule-based systems are easily extended by adding rules, an important feature since new special causes often arise.

To limit the number of diagnoses and increase the modularity of the system, the overall rule base has been partitioned by defining rulesets for each event type. Each ruleset is used to diagnose the special cause of variation that led to that event. Such modularity makes it easier to add knowledge when it becomes available.

Diagnosis is performed by applying the appropriate ruleset to a given event in a backward chaining fashion. That is, a list of hypothesized special causes is tested sequentially by the rules. Each rule in a ruleset is composed of a set of predicates which examine other data in the database to see if conditions correspond to a set of symptoms that have in the past been indicative of a particular special cause of variation.

An example rule, written in English, is as follows:

IF there was a cobble or sample taken on this coil.

AND there are thickness spikes in the body of the coil

AND one or more of those spikes is roughly at a distance in the coil that corresponds to the distance between the 96 inch reversing mill and the 80 inch continuous mill.

THEN diagnose this peak event as a cold spot on the slab due to the slab lying in contact with t he 96 inch workroll after it was backed out of the continuous mill.

In this rule, a cobble is a special situation in which tension is lost in the coil being rolled and the mill must be stopped. A sample is sometimes taken from a coil by stopping the mill and cutting the metal. Each of these pieces of information about the processing history of a particular coil is stored in the database. The rule above expresses some process knowledge related to the distance between successive rolling mills and how that might be related to the production of thickness spikes. If this set of data conditions obtain, then the rule concludes that the out-of-control peak in the signal was due to a particular special cause - a cause that is actually an operator error in handling the coil during a cobble or sample operation.

Multiple rules in the ruleset come into play in proving some of the conditions. For example, a separate rule does some data analysis to conclude that thickness spikes occurred in the body of the coil as opposed to the leading edge or the last few feet. When a diagnosis operation is performed on a set of events, the events are partitioned into diagnosed and undiagnosed sets. A report lists which causes could be diagnosed for each event. Multiple diagnoses are possible for one event. Further explanation of the diagnosis of a particular event can be requested. Figure 3 shows an English explanation printed by the program, of the diagnosis of a bandwidth-peak event on 12 Feb 88 at 6 p.m.

```
Event BANDWIDTH-PEAK at 79358 Lot Number = 68785] on 12 Feb 88 6:00pm.
 Diagnosis is (COLD-SPOT-FROM-96-DUE-TO-COBBLE-OR-SAMPLE).
      Report of Special Cause COLD-SPOT-FROM-96-DUE-TO-COBBLE-OR-SAMPLE
There was a 4-stand sample taken on this coil.
There are 3 spikes in the body of this coil.
  There is a gauge spike at 1832 ft. into coil of amplitude 2570 micro-inch.
  There is a gauge spike at 1811 ft. into coil of amplitude -1450 micro-inch.
  There is a gauge spike at 171 ft. into coil of amplitude 1589 micro-inch.
```

Figure 3. Printout of Program generated Explanation of Diagnosis

Undiagnosed Events: Using Process Knowledge to Discover Associated Variation

There will be new special causes that cannot be diagnosed with the current rules. The undiagnosed events that result must be analyzed to discover causes, and accompanying symptoms. A human expert would carry out such an investigation in several ways. A first step might be to examine other data collected, based on some model of what variations might have influenced the signal of interest. Out-of-control variations which occur at the same time as the signal variation of interest should be strong clues which may lead to a theory of what happened. These associations are typically augmented with information that may be unrecorded (operator observations, etc.).

We have developed a "discover" feature that uses a qualitative model of the process to look for associated symptoms in other signals. Figure 4 shows a greatly simplified association diagram or network which expresses the associations among the physically related variables that lead to variation in bandwidth.

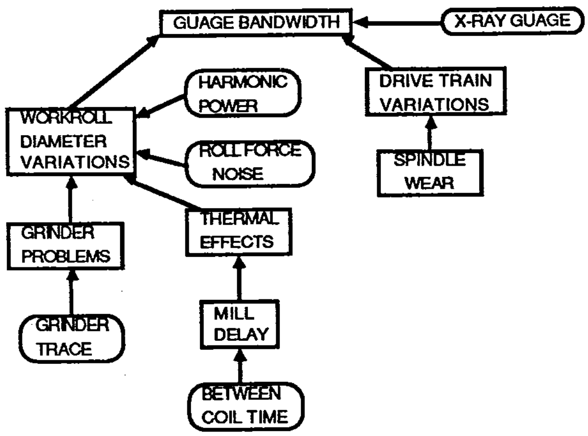

Figure 4. An Association Diagram for Bandwidth.

The internal representation of our association diagrams (also called an ANET) is a network of objects which are built using the FLAVORS programming system. There are two basic types of objects: measured quantities and derived quantities. Each "derived" object can contain links to measurements and other derived objects. Measurement objects contain a reference to a generator procedure which may be invoked to construct a set of signal features over a given interval of investigation. In Figure 4 the measured quantities are shown by oval symbols while the derived quantities are the rectangular boxes. The diagram suggests that workroll diameter variations influence bandwidth directly and are measured by both harmonic power and roll force noise.

The links between boxes can express more quantitative relationships. For example, one signal can influence another as a simple proportion. It could also have an integral relation where a spike in one signal leads to a step in another. Operators that express these relationships are carried on the link data structures in the network.

An ANET is a static, declarative representation of process knowledge. To be useful, it must be combined with a search procedure. For the purpose of discovering associated variations, a backward search through the network is appropriate. A backward search starts with the node representing the signal under investigation, i.e. the signal for which an event has been detected. A list of signal features is collected for the interval of investigation for all measurements of the starting node. These lists of signal features are then propagated recursively back through the network. When an association link is traversed, any operators attached to the link are used to transform the feature lists. When a node is reached which has measurements, feature lists for the measurements are generated and compared with the propagating lists. If a coincidence is found, an association is established and saved. The search is exhaustive and all associations which can be reached from the starting node are considered. The result of the search is a list of signals with one or more features that are coincident with some feature of the signal under investigation. Note that this result is close to the one desired when people study draftsman plots (a draftsman plot is a large, graphical covariance matrix, where each entry in the matrix is a scatterplot of the row vs. the column variable). However, cause-and-effect conclusions can generally be drawn more confidently from sets of signals which have been thus identified by an ANET than from draftsman plots for two reasons. First, draftman plots are limited by graphics device resolution to no more than twenty or so variables. Second, the ANET-based results contain domain-specific knowledge - consistently applied, whereas draftsman plots are ignorant of the domain.

The associated variations of an event can be viewed as a "signature" which characterizes the event. This signature can be used to investigate the nature of the event. The most obvious use of an event signature is to focus attention on the set of signals it reveals. It is often possible to gain insight into an event by scrutiny of simultaneous graphs of the set of signals in the event signature. A more intriguing possibility is the use of event signatures to classify (cluster) events. Classification of events in this manner can be either a manual or an automated procedure.

Development History

In early 1985 one of the authors (PLL) was involved in a new project at Alcoa Technical Center (ATC) which required the examination of large volumes of data from an instrumented rolling mill. The objective of this project was increased understanding of the rolling process and phenomena which impact product quality and consistency. It was discovered that a major impediment to this project was access and manipulation of the online data. The data were not organized in any rational manner and there were no software tools for effective interactive presentation of the data. It was decided that the importance of the project warranted the development of a software tool designed to support highly interactive access and manipulation of large volumes of data.

PROSAIC was developed on the Symbolics Lisp workstation. The choice of this platform was based on: (1) superiority of the software development environment, (2) need for a high level of hardware resources to support the proposed performance of the tool, (3) the intent to add automated reasoning capabilities. An initial prototype was built within about three months. This progress was due to the enlistment of the consulting services of BBN Laboratories (KRA), the use of a commercial software shell (KEE™), and the recycling of code from an existing application (CRH).

Initially, the system was installed and used at ATC. Development and use proceeded in parallel. The initial users of the system were ATC engineers working on the rolling process consistency project. This initial period of use resulted in many enhancements to the user interface predicated on user needs. In October, 1986 PROSAIC was installed at Alcoa's Tennessee Works.

The shift in usage from a research environment to a plant environment triggered many changes to PROSAIC. User support became a significant overhead issue. Previously ignored issues of interface robustness became critical. The initial implementation of the data system was found to be cumbersome and a complete reimplementation was necessary (DAS). As part of reimplementation of the data system, the frame based data dictionary was ported from KEE to an existing, in-house frame system (CRH). Considerable effort was expended toward rationalizing and generalizing the user interface (KRA). Much of the impetus for this work came from the plant users of the system. Their contributions included not only reports of system deficiencies but also suggestions for new tools and features which would aid them in their work.

Early in the development of PROSAIC it was realized that automated mechanisms for data exploration were highly desirable. The initial work on signal feature detection (PLL & MS) began shortly after the initial installation of PROSAIC at ATC. This early work resulted in some general concepts around what we now call special cause management (SCM) (Love & Simaan 1988). The press of events delayed extension of these ideas unitl September 1987 when the work described in this paper began.

The initial focus of the SCM work was to represent the physical causes of process variation as a diagram of relationships between signals (PLL & CRH). It was quickly apparent that this approach works well as a mechanism for discovery of new causes of process variation but that the more traditional, rule-based approach is appropriate for identifying well understood special causes. The rule-based system for diagnosing special causes was implemented with an existing, in-house rule system (CRH) together with an interface to the object oriented representation of events (PLL & CRH). The combination of these two approaches to special cause diagnosis and discovery, together with event archiving (CRH & DAS) and statistical refinements to signal feature detection (PLL, DEC, & APJ), constitute the framework for SCM.

Current Status and Future Directions

PROSAIC has been in use in a plant setting for well over two years. During this time the system has proved to be useful for many purposes, some of which are listed below:

- Investigation of out-of-control bandwidth events, leading to procedural changes and a reduction of bandwidth exceptional coils. Knowledge gained in this study has been implemented in SCM rules.
- Discovery of work roll grinder problems leading to excessive bandwidth events. Investigation of these events lead to a better understanding of the relationship between roll grinding and bandwidth.
- A better understanding of the relationship between product metallurgical properties and process parameters.
- A better understanding of limitations intrinsic to the current rolling equipment. This information was used to plan for a major mill modernization.
- Monitoring and evaluation during the installation of new processes and comparison of trial runs to historical data.
- Process and equipment troubleshooting.
- Investigation of customer complaints.

The accomplishments listed above were achieved using the manual data exploration features of PROSAIC. Much of the knowledge gained during these activities has been coded in the newer SCM features of the system. Much work remains before the SCM features of PROSAIC begin to perform at a level sufficient to replace manual data exploration activities. Only a small part of the rolling process - bandwidth anomalies- is currently embodied in the SCM tools. The system does a reasonable job of identifying special causes for bandwidth events, although approximately 30 to 40% of the detected bandwidth events still go undiagnosed.

The PROSAIC system continues to be actively developed. The system has recently been ported to the MacIvory™ microcomputer with the objective of providing a low cost delivery platform which can be spread to multiple sites. The data system is being enhanced and generalized to enable application of PROSAIC to domains other than the aluminum rolling process. Finally, refinement and extension of the SCM tools discussed in this paper are a major priority for future work.

Acknowledgements

The authors wish to acknowledge the significant contribution of Mark Pate to the development of PROSAIC. Mark, who is an engineer at Alcoa's Tennessee Works, has been the major user of the system since its installation. He has provided invaluable input as a computer literate and enthusiastic user, and has provided the system developers with numerous ideas for system enhancements.

References

H.M. Wadsworth, K.S. Stephens and A. B. Godfrey 1986. Modern Methods For Quality Control And Improvement, New York:Wiley.

W. A. Shewhart 1926 "Quality Control Charts," Bell System Technical Journal, pp. 593-603, October.

W.A. Shewhart 1931. Economic Control Of Quality And Manufactured Products, New York: Van Nostrand.

H.D. Maragah and W.H. Woodall 1988. "The Effect of Autocorrelation on the Shewhart Quality Control Chart with Supplementary Run Rules," paper presented at the Joint Statistical Meetings, ASA.

P. L. Love and M. Simaan 1988. "Automatic Recognition of Primitive Signals in Manufacturing Process Signals," Pattern Recognition, Vol 21, No. 4.

G. Stockman, L. N. Kanal and M. G. Kyle 1976. "Structural Recognition of Cartoid Pulse Waves Using A General Waveform Parsing System," Commu. Ass. Comput. Mach., Vol 19, pp. 690-695.

K. P. Birman 1982. " Rule-Based Learning For More Accurate ECG Analysis," IEEE Trans. Pattern Anal. Mach. Intell., Vol PAMI-4, pp. 369-379.

K. R. Anderson 1982. "Syntactic Analysis Of Seismic Waveforms Using Augmented Transition Network Grammars, Geoexploration, Vol 20, pp. 161-182.

K. S. Fu 1974. Syntactical Methods In Pattern Recognition, New York: Academic Press.

J. W. Tukey 1974. "Non-linear (nonsuperposable) Method For Smoothing Data," Proceedings of the 1974 EASCON Conference.

D. Weinreb and D. Moon 1981. The Lisp Machine Manual, Cambridge: MIT Press.

F. Hayes-Roth, D. A. Waterman and D. B. Lenant 1983. Building Expert Systems, Reading: Addison-Wesley.

B. G. Buchanan and E. H. Shortliffe 1983. Rule-Based Expert Systems: The Mycin Experiments of the Heuristic Programming Project, Reading: Addison-Wesley.