INFORMATION EVOLUTION
Reliable, Affordable, Human Bandwidth

Poka Yoke Processes and Database Quality

Table of Contents

White Paper

By Matt Manning
President
Information Evolution Inc.

Last Updated: April 2008

Overview

The Japanese concept of poka yoke (mistake-proofing) was developed as an inexpensive and very effective way for improving the results of manufacturing assembly processes. The principle is sublime in its simplicity: Instead of only doing random sampling of products after they are created to find errors and then exhorting employees to "do better" via various "carrots" and "sticks," it is better to reengineer the process to prevent an error from occurring in the first place.

Typical manufacturing examples are cases where a stage in a process cannot commence at all until the previous stage in the process is proven to have been done correctly. So, for example, a machine that screws in a particular screw at a particular stage is designed so it can *only* screw in the right screw (one with a corresponding head) or the second stage can't begin (i.e., the power won't turn on) if only 3 of 4 of the four screws required aren't present.

Applying Poka Yoke to the Information Industry

This principle is not just applicable to manufacturing a physical product, but also in creating information products as well. As any editor knows, one of the bedrock principles of effective copy editing is that same person doesn't check the work they have done themselves. But beyond that, how much thought do we typically give to the processes that drive the quality of our products? And does outsourcing those processes relieve us of the responsibility for knowing specifically "how the sausage is made" and trying to improve those processes?

Savvy publishers are hands-on about the many important decisions that make the difference between passable quality and eye-opening value. So what types of data-related editorial processes could benefit from poka yoke-type reengineering?

Variance Identification

Every field has one or more "size" or "frequency" parameters. Size measures include the length of the value in a field (the number of characters in it) or the numerical value of a field. Frequency measures indicate how often the field contains a particular value.

For lack of a better phrase, "variance identification"—checking for records that do not conform to the size or frequency parameters defined for a given field—is the most obvious area for publishers to concentrate on. Rather than just looking for variances during random checks or after the

"Variance identification"—checking for records that do not conform to the size or frequency parameters defined for a given field—is the most obvious area for publishers to concentrate on.

production process, quality-conscious publishers have built-in validation checks for variances, the most common being "null" value prompts ("Field X has a value of zero or is blank. Is this correct?") and "100% of allowable length" prompts to ensure that a record isn't entered into the database potentially having a truncation issue.

Anomalous Errors

Flagging "anomalous" values in mid-stream is also possible ("Are you sure the value of 25 employees is correct? The company has annual revenues of $1.2 billion."), but rarely done, despite how relatively easy it is to do. Does a 35-year-old person have a 30-year-old child? Does someone live in New York and have a California phone number? The number of double-checks can go on and on, but adding additional checks is usually a matter of a few minutes of development time, so why not cover as many bases as possible?

It is true that most major "anomaly" errors are found by doing sorts by every relevant field at the end of the initial editorial process and examining the high and low values, which is a very valuable and important process. It is, however, far more efficient to flag anomalies while a researcher is in the middle of working on a record than after the fact. After all, an error cannot ever reach the eyes of an end-user if it is never saved in the database in the first place. Once it's in the database, maybe it will be corrected, or maybe not.

A series of a few dozen triggers built into a typical database updating process can catch most errors in real-time and dramatically improve product quality.

A series of a few dozen triggers built into a typical database updating process can catch most errors in real-time and dramatically improve product quality. Once in place, these processes are very easy to refine. They also have the added advantages of shortening the editorial cycle, requiring less domain knowledge on the part of the front-line researcher by embedding "intelligence" into the process (which is defined by the domain expert), and reducing costs.

Embedded Style Guides

Another tool that could be considered similar to poka yoke in principle is that of embedded style guides. Embedded checks that trigger pop-up dialog boxes when a non-approved term appears are fairly common ("The approved abbreviation for 'Street' is 'St.' Do you want to accept the change?"), but sophisticated tools that ensure use of Chicago Manual of Style guidelines for large text blocks and enforce tricky rules about things like the use of a passive versus an active voice are now available, and can eliminate the need for a manual editorial check.

About the Author

Matt Manning is the president of Information Evolution, Inc., a firm that designs and implements efficient research and editorial processes for content companies.

About Information Evolution, Inc.

IEI provides human resource and technology services to companies, primarily in publishing or related industries, that manage large databases in real time. For more information, call (512) 650-1111 or visit www.informationevolution.com.