



Principles of Adaptive Technology

Believing The Last Liar

Accurate data and error prevention are key components in the foundation of a well-run distribution center. Multiple resources are allocated to assure that the electronic information that supports the distribution center is error-free. Unfortunately, regardless of all the efforts, sooner or later inaccuracies will happen; for instance, operators will make mistakes, electronic data from vendors will not be fully accurate, equipment will not work as directed, etc. For all these reasons, error recovery must be considered as important as error prevention in the operation of a distribution center. This paper addresses error recovery in distribution centers.

One Plus One Is Normally Two

In the mathematician's world as well as in most information systems, one plus one is always equal to two. For a distribution center operation, a more realistic statement is that one plus one is normally two. If the data were always precise, one plus one would be always two; however, when data could be an imperfect reflection of reality, the best bet is to only assume that one plus one is two most of the time.

System conditions in a distribution center are constantly changing. Information systems use multiple sensors – for instance operators, equipment, and other data sources – to learn about the changes and update the data to reflect the new reality. It is expected the information system will always have up to date accurate image of “what is”. The problem is that those sensors do not always report the truth to the information system. From time to time they lie and nobody, including the information system, knows when that will happen. Now, an element that is constantly generating inaccurate data is a faulty element that needs to be replaced. However, non-faulty elements also generate inaccurate data – for instance, the best picker could pick the wrong item from time to time – and replacing those elements is not going to prevent the errors. Errors are a fact of life that cannot be avoided. A good information system must have error recovery features to deal with these errors.

Good Liars and Bad Liars

Ultimately each piece of data received by an information system originates at a source that is known to, upon occasion, be inaccurate; hence the source is a “liar”. There are bad liars, good liars and excellent liars. Bad liars lie all the time and are easily identified. We really never believe a bad liar. The data provided by a good liar is normally assumed to be accurate. An interesting observation is that the very best liar turns out to be the one that most rarely lies. When an excellent liar provides inaccurate data, we assume it to be correct.

Believing the Last Liar

An approach that has proven to handle the unavoidable errors very well is based on the following principles:

- 1) In spite of the fact that sensors feed the information system mainly accurate data, there is always the potential for the data to be faulty.
- 2) When possible, if the sensor happens to be a person, provide output indicating that there is an inconsistency. Then accept the resulting response.

- 3) When the information system receives data from a sensor that is inconsistent with data previously received, the newer data is more likely to be correct than the old data.

This approach is known as *"Believing the Last Liar"*. The alternate approach, the approach that is used in most distribution center information systems, is one that is best termed "believing the first liar". This approach is easily identified, when inconsistent situations are encountered, reality (the "what is") is forced to match the image of reality held by the information system. Sure the information system may have means to correct the image, but correction takes the form of the information system inherently believing the initial lie.

Error Recovery – Handling Exceptions

Believing the last liar is just the first step of the error recovery procedure. Error recovery is in fact exception handling. If we were to have no errors, we would have no exceptions. New unplanned actions are always required handle exceptions. Streamlining the handling of exceptions is a great start. That allows the focus of operation to center on where we are to go from here rather than how we arrived in a particular situation. Too many times exception handling processes seem to focus on "killing the messenger". We have actually heard information system programmers state "we need to make exception handling as difficult as possible to discourage mistakes". From our perspective, individuals with that point of view are consuming way too much of one of the earth's most valuable resources – namely air. Difficult exception handling, i.e. the discovery of an inconsistent situation, will never discourage mistakes; it will only discourage the correction of previous mistakes.

Examples

Example 1:

The information system has carton "X" as part of pallet "Y" in pallet rack location "Z". The information system has not issued any transaction to move that carton. Suddenly, carton "X" is scanned on a conveyor. As the scanner is the last liar, the carton is most likely to be on the conveyor rather than in the pallet rack. So, the information system relocates carton "X" to the conveyor where it was scanned and remove the carton from the pallet "Y". The next issue is to decide what to do with the carton. If the contents of the carton are needed to fill an order, that is likely to be the best destination for the carton. If the carton needs to be placed back in storage it would be easier to place it in a carton rack than trying to marry it again with its original pallet.

Example 2:

A picker is walking along a pick loop picking items according to transactions issued to him through a RF unit. The transactions are issued according to a pick location map and the last pick transaction (optimized route). Suddenly the picker scans a location far from the loop segment where he was supposed to be (based on the previous transaction). So, the information system relocates the picker in front of the scanned location and re-optimizes the issued transactions from that new location.

Conclusion

No one argues that when the information system reality and the physical reality do not match there is an error. When a carton is found in a location different than the one reported in the information system some may say that the carton is in the wrong location, while others say that the information system has the carton in the wrong location. In the end, the issue of which is wrong is irrelevant. If you usually find it easier to force physical reality to match that of the information system, you may be being punished for an earlier mistake. You most likely have an information system that inherently believes the first liar. The most important issue is to decide what is the best action to take given the current conditions. Adaptive systems inherently believe the last liar. All actions are based on the most current conditions. **Let's not shoot the messenger!**