**MANAGE YOUR RISKS AND THREATS BY MANAGING YOUR DATA**

## Introduction

The corrosion department was probably one of the first groups, *if not the first* within the pipeline maintenance crews to develop and use handwritten databases to compile data collected during field surveys. They would use these handwritten databases to analyze, and make several determinations; from checking the protection status of their systems, forecasting future needs, and even to demonstrate why improvements may be needed. Also, these databases would be used to assist in developing annual budgets, upgrades and repair estimates.

The more data collected and used, the more sophisticated the databases became. Soon, it became obvious pipe-to-soil potentials was not the only data needed to be recorded, it was soon realized that the same pipe-to-soil potential data would be more meaningful if coupled with other metadata taken at the same time. Over time, a lot of data and *related* metadata has been compiled, and not only by the corrosion department, but other departments as well. Today, it seems the pipeline industry (*not unlike other industries*) is practically over-whelmed with good meaningful data – *not to mention the potentially great data possibly lost over time*.

## The Computer

Pipelines have been used for transportation of hydrocarbons since the late 1800’s and early 1900’s but the thought of collecting and maintaining pipeline operating and maintenance data had not really begun in earnest until the 1950’s or so. Some of this earlier data can still be found in boxes of dusty folders stored in a storage Wearhouse somewhere, but sadly a lot of this data has been lost to the ravage of time. Today the task of managing and storing data is helped along in large part by computers and computer programs, making the task much quicker and much more efficient and error-free.

It was not too many decades ago when a rather large room would be designed and dedicated to house just one computer. These earlier computers not only required a lot of room, but also needed a designated cooling system to prevent the computers from overheating during operation. These earlier computers, *probably needless to say*, were not real user-friendly. As large as the earlier computers were, they were generally only programed to run one task, whereas computers today are not only programed to operate several programs but multitask several programs all at one time (*mostly thanks to inroads made in miniaturizing everything during the early space program of the 1960’s*). Today the same capabilities of these earlier computers are provided in computers that are small enough to be hand-held, and can do much more than their early predecessors. It seems there is at least one computer program involved in almost all we do today, and in some cases through initial design of the programs and systems, computers and their programs are actually needed. As an example, in some of the aircraft today, should an aircraft’s computer fail, it would be very difficult, and probably *even impossible* to pilot manually.

Today computers have made such pipeline processes as engineering, design, operations and even maintenance much quicker and much more efficient. The computer’s first real inroads to the pipeline industry came as a need in both engineering and design, but today some pipeline systems are controlled, monitored and operated, for the most part by computers. Today most of what we do, *so effortlessly*, may have been very difficult to achieve in a timely manner before the use of the computers.

## Pipeline Operations and Computers

Some of the processes and tasks of operating a pipeline system today using the computer were not even considered in the pipeline industry’s earlier history, simply because they were not feasible or even possible without the use of a computer system. The *computational pipeline monitoring* (CPM) systems can be found in use today in almost all pipeline systems in some fashion or another. Some of the earlier CPM systems seem rather simple by today’s standards, but they were all basically designed and installed simply to better detect and alarm the operators of possible leaks and other abnormalities, allowing them to more quickly get a situation or incident under control. The earlier CPM systems were designed to do little more than monitor the pipeline’s input and output pressure, temperature and flow rates, and should it detect any difference in input and output, it would alarm the operator of a detection of possible problems. Today these computer systems have since gotten much more sophisticated and now play a much bigger role in all facet of pipeline operations.

Probably developed because of the CPM, or perhaps because of nothing more than just a natural progression of things, the pipeline owners and operators have raised the bar, *or up the ante as it were*, by installing *supervisory control and data acquisition* (SCADA) systems. Today with the ever-developing use of the computer, several pipeline systems can virtually be monitored and controlled from one central remote location through the SCADA system. As the computers have become much smaller and more productive, so has the peripherals and programming.

By doing what is expected of various computer systems, and their programing, a lot of data is acquired and stored. Today, one of the biggest reason for data collection and storage is because most programs, such as the SCADA system need to gather and store data while making certain decisions, reacting automatically to operation abnormalities or to maintain control within the pipeline’s designed operating parameters. The pipeline industry quickly began to realize that some of the data collected and stored by each computer’s program could be very useful in answering various questions that may arise from a specific span of time during the pipeline’s past operation, such as during an emergency, or during an abnormal operating condition. It now has become a normal mode of operation to query a program system’s database from *time-to-time* to answer many questions, or maybe just to simply satisfy one’s own curiosity as to how well a recent alteration to the pipeline system is performing.

## Database Access

The pipeline industry is in a mode of sorts where there exists, for each pipeline system several different computer systems and programs, and in most cases managing their independent area of concern and having their own separate databases. As the needs change through design and/or regulations we have become more aware of the need and conveniences of accessing these databases to accommodate the many changes in operations, design and other facets of owning or operating a pipeline. In the past, when fewer and simpler programs were used, it had not been too much of a problem having and using a separate database for each program, because of information, whenever needed, one would just go to that particular program, retrieve whatever information that database had available, and go on to the another database if more information is needed, *and hoped to be available (i.e. general data, drawing storage, metering data, corrosion prevention data, etc.)*. It has quickly become the norm that more than one database access for information is needed to satisfy a particular operating issue or situation. Because a lot of the databases were originally designed to merely satisfy a particular program’s needs, sometimes not all data retrieved from multiple databases seems to merge well together in a real meaningful manner, and sometimes it become quite a task just to get all that is needed, and hopefully nothing important falls off the table. This is where a real *database managing program* (DBMP) begins its introduction into the history of computers and computer programming.

## The Database Managing Program (DBMP)

Some pipeline owners and operators are already looking into buying or developing their own DBMP mainly for the reasons previously mentioned. A good DBMP should be multifaceted and basically consist of a standalone foundation program that has the capability of being customized and integrated through programing to meet the needs of whatever processes or tasks the DBMP is going to be used for.

While a database does basically nothing more than just store and retrieve data, a good DBMP should not only remember where everything exists, but also remember how to quickly retrieve it, when needed. A DBMP will need to be designed to manage multiple databases, while leaving them, and their data content intact and unchanged. A DBMP needs to be able to copy and merge together all, or some information from all available databases into one temporary database to work from. This will keep the DBMP from going back and forth among all the various databases, thus making it’s operation much quicker. A good DBMP should be able to help in making almost all decisions it is programed for, and even offer up suggested options with possible end results for resolving many questions and requirements that come with managing a large ongoing process such as a pipeline’s integrity management program (IMP).

A good DBMP will need to be programable to accept any, and all forms of data such as; handwritten, PDF files, photographs, drawings, quantitative values, etc. A DBMP should have the ability to not only access and retrieve data from other databases, it should also be programable to evaluate the data as it receives it, and ensure it understands all it needs to know in order to store it properly, and should check against all other similar data that already exists for only this particular pipeline system or segment.

Sometimes while data is being entered, it appears to be a duplication of what already exists in another form or may not merge well with what is already in use. Should this happen, the DBMP should prompt the computer program operator to check it and make any necessary alterations to ensure the data is good and stored properly. For historical purposes, the DBMP should not over-write any data. The DBMP should simply flag suspected issues, leaving any resulting or necessary data alterations to the pipeline operator. A good DBMP will generate a detailed process log, so that it can be reviewed to check where all the data used is derived from and how it was used during a programed process. This log will allow the operator to correct anything that may have been incorrectly performed or interpreted by the DBMP during its processes.

## The Many Uses of a Good DBMP

Most of the following discussion will be centered on typical IMP processes, and just as a good example of how well a DBMP should be able to assists in improving the performance of any IMP and its end results. A DBMP should have the capabilities to offer much more than just managing an IMP process, it should also be able to manage other large processes as well, such as *project management, operator qualification, public awareness, etc.* After all, needed information for all of these processes should be derived from the same multiple databases.

Most pipeline operators’ IMP, *just by their programing nature*, have become massive database users. One important requirement for a good IMP to perform properly, with good results, is having all available information quickly within an operator’s use, no matter what the discipline is in. The IMP regulations have been in place now for many years, and because of this, the pipeline operator has amassed heaps of data from many different databases and should use all that is available. Within these pre-existing databases, there lies some very valuable and much needed information. Some of this data has probably been available for years and probably forgotten about to some degree, but just the same, provides very important information for so many processes within the IMP. For an IMP to be successful it needs to be ever-aware of all historical data, no matter how insignificant it may seem at the time it was first entered, and as we’ll see, should be data considered during all phases, or implementation of the many IMP processes. As is the nature of databases generated by various different programs, all data collected and stored, *no matter the collection process or format used*, is associated with a particular pipeline system, or a segment within a pipeline system. A DBMP should be programmable to understand these and many more associations by knowing just where to find data when needed – again, no matter how insignificant that data may have seemed initially. This DBMP should be discriminate when considering available data, if data is available for whatever reason; the DBMP should automatically bring it to the forefront offering it up as a possible consideration for use, no matter what IMP process is being implemented or already in use.

## Implementing an IMP Process

For the benefit of this IMP discussion we will use a mock project call the *Recoat Project* performed sometime in the past. This project takes place some 14 years earlier on a 30-year-old pipeline system. We will use this mock project, and all its available data as a single maintenance incident that we will carry throughout various examples of how a DBMP should work, while explaining and highlighting the many useful program features, expectations and uses that should be available in a good DBMP. Also, in doing this it is hoped that the reader will better understand how important one item from some otherwise 14 years old, seemingly obscured historical data can be used with other relevant data repeatedly as a very useful means for something as involved and massive as a good IMP, while giving some insight into how a good DBMP should be expected to work. This writing will also highlight other features that should also be available and should be considered.

A good example of how a DBMP should work within many aspects of operating a pipeline, just by being always available, and waiting for the go-ahead to run. Take the following common example; the DBMP has in one of its many database records of a 1,500 feet segment of pipeline being recoated, and within another database it is also noted that its cathodic protection (CP) system’s protective *current output* for the same general area was cut-back or reduced soon afterwards. More than likely as a direct result of the recoating project. A good maintenance record at the time, but information not thought of or used much since, and probably thought to serve as a very good piece of historical data to have for various reasons at the time it was recorded. While trying to determine threats for this pipeline during an IMP risk analysis process, it becomes even more important and becomes a consideration to the DBMP, rather than just a maintenance record. The DBMP should be aware that this information is available, because it should be within its basic programing to always consider all available databases during any process it is asked to perform.

It is the job of the DBMPis to be aware of this data through its continued awareness and recognition programing that a lot of good information exists from many records for various reasons. One of the items the DBMP will find noteworthy and possibly important as reference data to consider, is the *Recoat Project* did not include any pipe replacement or repairs. According to the recorded data the external wall loss was found to be minor overall, but covered a large area, leaving more than enough wall thickness to continue operating the pipeline safely. But again, the wall loss did cover a large area as general corrosion, and the *Recoat Project* was quite an undertaking. It is also noted in this particular maintenance, or data report that the root-cause was determined to be the initial coating was poorly applied, eventually leaving too much of the pipe’s exterior surface exposed to a corrosive environment, and no other threat contribution was noted. The CP system was in play but was being operated at a level that if left uncorrected may have later caused more serious corrosion problems and possibly eventual failure. Too much bare surface area was exposed and was nearing the maximum limits of the CP’s initial output design. Before the coating repair, the CP maintenance records indicate that even with the CP system’s protective current at an elevated protection level, the *pipe-to-soil* potential readings were not what was required to fully protect the pipeline. When completed, and the backfill had settled, the *Recoat Project* appeared to be a success after follow-up investigations were performed. All indications were that the CP system’s *pipe-to-soil* potential readings had risen considerably, indicating the protective current could now be reduced. All the good work and good record keeping was probably initially recorded for prosperity, and thought to be nothing more than simply a matter of historical information.

## Risk Analysis and the DBMP

The DBMP, through its programing shouldalways be aware that risk analysis consideration exists, and will bring data like the *Recoat Project*, and others like it to the forefront for consideration, no matter where the data is stored and no matter the format it was stored in. Once the risk analysis process is implemented and is made aware of all necessary records, including relative details that the DBMP has extracted. The risk analysis results will be much more exact; a structural integrity threat that may have gone unnoticed will now be elevated to a higher level of consideration. After all, if a pipeline coating has already failed in one area then the remaining initial, plant-applied coating should be suspect elsewhere, and should be considered as a viable integrity threat. A good DBMP will make this bit of information that may have otherwise gone unnoticed, as being relevant to the risk analysis process.

Most IMPs have their own risk analysis program to calculate risk factors, and like many programs, the risk analysis program will need certain data to generate a risk factor. The risk factor will then be use within the IMP to make calculations and decisions needed by other IMP processes to continuously reduce this factor to its lowest possible value. A good DBMP can be programed to work together with the risk analysis program by locating and inputting the data the risk analysis program needs to calculate a current risk factor. Being able to do this, the DBMP could generate possible preventative and mitigative measure scenarios and provide results as to how much each would possibly affect the risk factor. A very good example of how the databases generated for the mock *Recoat Project* now takes on a whole new level of importance, thanks to the use of a good DBMP.

## Integrity Assessment Method and Remediation

The *Recoat Project* again comes into play while trying to determine an integrity assessment method and making determination for all possible remediation considerations. Thanks largely to the work of a DBMP during the risk analysis process; we now know external corrosion, due to coating deterioration noted in records of the *Recoat Project* should be considered a real threat. Because of a history of wall loss due to poor coating, the decision is made to use an inline inspection (ILI) tool as the method for integrity assessment. As the results of the tool run are analyzed, the data is then imported by the DBMP into its database for storage, retrieval and further analysis. While the DBMP considers the anomalies, as its programing dictates, and begins to prepare the dig sheets for remediation as required by the appropriate regulations. It should be noted that s DBMP will be programed to be aware of where the high consequence areas (HCA) are and uses this information to correctly prioritize the digs and repairs. To continue with our *Recoat Project* example, let’s say there are a total of 550 anomalies in need of attention, and are grouped in 27 general areas, with only 15 of these found within HCA. The DBMP will not have forgotten about the historical *Recoat Project* record and knows that 5 of the areas identified are called out as areas of concern because of their HCA location and are included as anomalies in need of repair. This recognition is added to its report, and suggests that these may have already been taken care of some 14 years earlier by the *Recoat Project* – remember no repairs or pipe replacement had occurred to restore any wall loss, so these could be the very same anomalies found at that earlier time. As it turns out the *Recoat Project’s* new coating was still in very good shape as verified by a few pipeline exposure excavations, and the wall loss depths that were called out by the ILI tool were, indeed preexisting, and with that bit of information, more time and effort would be better spent on the remaining 10 areas, of which were really of lesser consequence in this particular case.

## Preventative and Mitigative Measures (P&MM)

Remediation is complete, and a reassessment of risk is again performed. The overall risk value for the pipeline system has been reduced, and because not all the existing coating was replaced during the *Recoat Project*, it is now time to use our DBMP to determine any P&MM that will appropriately reduce the risk value even more. By accessing information from its database and all others available to it, the DBMP should be able to offer up possible P&MM that are available and will even offer an opportunity to provide possible effects each proposed measure will have on the risk values, should this P&MM measure be implement. If it has in its programming, and has access to necessary financial information, it can also provide an estimate for cost and any risk reduction realized for each possible P&MM considered.

## IMP Self Evaluation

With the DBMP involvement in all aspects of the IMP it will have all that is needed to perform or implement a very detailed IMP self-evaluation. A good DBMP should have every capability and available data for evaluating how well all IMP processes have performed, and in doing so, also produce records and demonstrations in any format a pipeline operator could think of to prove or show ongoing progress and any cost savings for both maintenance and operation of an overall pipeline system. Should IMP personnel be at a moment’s loss for a good demonstration.

## Conclusion

What better way to prove a theory than to do it with data? **Data is basically** Information in raw or unorganized form (*such as: alphabets, numbers, or symbols*) that refers to or represents conditions, ideas, or objects. Data is limitless and can be found everywhere and in many forms. A database will generally display data in the form of symbols or signals that are input, stored, and processed for output as usable information.

All data at some point gets updated, expanded and deleted as new information is continuously being updated or added-to. Data should be complete, and secure from alteration, inadvertent erasures or for whatever reason, just being completely lost all together. Back-up data needs to always be exact. Many activities concerning data needs to be documented and logged at times of acquiring. From the moment data is collected, used or added-to, there needs to be records that are considered original records, true copies, or other accurate reproductions of the original records. Data needs to be attributable, legible, concurrently recorded with all other relative data.

To be considered complete and meaningful data when derived from field, readings needs to be recorded with complete supporting information (*metadata*), and information as to the manner the said data was obtained.

Managing and using data is more than just downloading data into a database and querying it occasionally. Today’s computers have become great tools for doing almost anything, and in using the best available programs just makes the computer an even better tool. Granted, all the while, making the computer and its programs or software even more of a necessity. Hopefully this writing has given some insight into how all of this is possible. The many capabilities of a really good DBMP have only been touched-on here for introduction purposes, *with the surface only scratched*, but it is hoped this has instilled enough interest and consideration for acquiring a good innovative and powerful DBMP (database management program). In today’s environment, it’s almost a real “*must*”.