How to Make Useful and Cost Effective Application Software

=?

The Challenge of Sharing More than Information

Abstract

This paper discuss some of the consequences of the development of the information technology (IT) over the past 10-15 years, and why hydraulic models to some extend have suffered from this development. The future will continue to change the way models are used in the industry. The models will change from being almost synonymous with applications to be components in the IT infrastructure utilized in many different applications - and by users ranging from modeling experts to non-technical people with no understanding of the physical process of moving fluids in pipelines.

Introduction

Looking back some 15-20 years most pipelines were operated using pen and paper, telephone and fax, and computers primarily in four distinct and *independent* areas

- to design pipelines
- to analyze and plan operation,
- to acquire real time process data and remotely control the process (SCADA),
- to manage finances and accounting

What has changed since then since IT has become so important in the pipeline business? Could we still operate pipelines without the support of IT? And if not – what do we really need IT for?

The key to an answer to these questions is the dynamics of the pipeline, i.e. the expectation that the physical pipeline as well as the business can change continuously, and that decisions are required all the time in order to keep up with the ever increasing demand for flexibility.

Not too many years ago, the pipeline was very much in charge of the decision on how to operate. Plans could be made for month and years. Now the market expects that plans can be changed by the hour, and that the transportation cost is constantly reduced without compromising the security of supply.

The operation of the pipeline has always been a real time task, while planning and accounting is moving from a very much asynchronous 'back office' process to a front –end real time process. Customers of the pipeline want flexibility which means shorter planning horizons and the pipeline company want faster allocation as a basis for faster billing.

But it is not only shorter turn-around times – it is also a question of better utilization of the physical and human resources, and a larger focus on security of supply and protection of the environment.

This paper is an attempt to summarize the status after 10-15 years with a tremendous growth in the information technology and a simultaneous change in the way pipeline companies are operated. In some way, we have to restructure the way the IT is used in the pipeline industry – especially the way we use pipeline simulation. Quite unfair, pipeline simulation has got a negative image in the industry – mainly among non-technical staff as attempts have been made to justify investments in models with applications outside their traditional environment among engineers.

This is not to say that nothing has happened in the last years – but there is no doubt that some revisions can revive the pipeline simulation as a necessary tool in the industry.

The author's entry to this paper is that it is not the models to blame for their sometimes bad reputation. It is rather the application of the models outside their traditional habitat.

A discussion of hydraulic models is also a discussion of their applications, the users, and the IT infrastructure connecting all of this. A model is no longer associated with a single application, user and computer. It is a component in a larger IT context and deserves to be treated as such by all parties involved.

With easier access to information and a need for faster decisions, it has been tempting to try to open the access to the traditional hydraulic models that previously were reserved for very specific applications by a small group of users with an engineering background.

It has been equally tempting to open the access to the traditional modeling of business processes to allow engineers and operators to consider the economical impact of the technical decisions.

While all departments could agree on these ideas, many forgot to discuss the different meanings of the word model and the basic skills required to use a model.

What Is A Model?

To most people, a (computer-based) model is something with equations, constraints, logics, variables and parameters used to calculate some useful information based on known but not directly applicable information. In the broadest sense, a model can be used to calculate anything from the expected turn-over and profit in the company, and the accumulated flow to a customer to the pressure and flow distribution in a pipeline.

Individuals often think that if they master one model it should not be too difficult to master another. Experience clearly shows that a model is only part of the solution – the skills of the user(s) are at least as important. If the users are not familiar with the concepts behind, the data processing inside and the interpretation of the model results, the model can easily turn out to be worthless.

We have already accepted that in order to share information, we need an IT infra-structure as much as we need roads and airlines. We have also accepted, that in order to use the Intranet and Internet we need to use common transportation vehicles in form of communication standards and shared software from word processors and spread sheets to browsers. We are also slowly accepting that the content of the vehicles has to be standardized to some extend as it appears from amongst others the PSIG standardization of data structures for pipelines. But in many respects we are still struggling with extremely different modeling concepts and differences in user skills.

The following simple example serves to illustrate the last two dilemmas. Consider a simple steady-state model of a single liquid pipe:

$$P_u - P_d = 2 f \rho_o^2 L Q^2 / (\rho A^2 d)$$

The *business developer* applies this model in some form to evaluate the economical feasibility of building a new pipeline (how much compression do I need to move Q over a distance L for different pipe diameters?).

The *hydraulic design engineer* uses this model to calculate a minimum pipe diameter, d, to obtain a specified design capacity, Q, for an available pressure drop, $P_u - P_d$.

The *facility planner* applies this equation to calculate the available flow capacity for the available pressures taking into account non-availability of devices due to maintenance or failure.

The *pipeline operator* uses the equation to evaluate need for pigging (reduced efficiency) and for calculation of required inlet pressure to get the desired flow.

The *pipeline scheduler* may not use the equation directly (he has no up-to-date feeling for the available pressure) but he would like to know the current state as basis for his transport planning.

Neither does the *pipeline accountant*, who is only interested in the quantities entering and leaving the pipeline over a period of time. However if the accountant want to analyse the relation between required pumping/compression costs and throughput in order to analyze the profitability, he will need this equation as part of his cost model.

For such a simple ‘model’, most people understand how to get the information needed, and they also understand that use of the model depends on the availability of in this case 7 numbers to calculate one. They generally also appreciate that in order to get reliable input, they often have to consult different experts in the company.

This example illustrates how the same equation can be used for many different applications by many different users, but also that optimal use requires cooperation between different people who individually applies the model for different purposes.

When we look at more complex/realistic models, this is no longer as obvious as here, and often a group of users believe that they can use the same model and that the necessary input always is available. This is understandable, but unfortunately, you have often spend a significant amount of resources before it is clear that either the modeling tool does not readily support the desired applications or the necessary input cannot be made available with the necessary accuracy. Or it turns out that the time it takes to get an answer is too long – especially considering the quality of the input and the requirements for accuracy of the results.

While the equation above easily can be solved for any parameter or variable, this is generally not the case for just a little more complex models. One clearly documented example is the API standard, API MPMS Chapter 11.1-Volume VIII , for correction of volume and density, which in some form or another is used in almost any pipeline context from design to operation and accounting. This standard explains in 200+ pages all about unit conversion, iterative calculation schemes etc. for something as ‘simple’ as conversion of volume and density from one unit system to another and from one reference state (P,T) to another.

> The following generic definition of the term *model* is suggested:
>
> “A model is a collection of equations, constraints, logics, parameters and variables, and a solution algorithm providing the values of specified variables (output) for specified values of other variables (input) and parameters”.

With this definition, the equation above is not a model in itself but a part of 8 different models – perhaps confusing to many, but actually correct when it comes to implementing computerized models. Although the 8 models has a lot in common, it is very important to focus on their differences because they have a severe impact on the application of the models, the input and the solution algorithm required.

What Is An Application?

As illustrated above there may be many potential applications of one model, and one application can also require several different models.

One application represents one users use of a set of input data, and one or more models as required to generate exactly the output needed in his work process, and exactly in a form suitable for this purpose.

In this context it is important to emphasize that output is not only a question of some numbers presented in graphical or tabular form. It is also a question of trust in the numbers presented. The user has to be able to and have the tools to validate the output.

If the user is requested to provide or select input for his application, he shall also be capable of evaluating the data quality and the impact of bad input on the output.

A model is not an application in itself. It is made an application by adding configuration tools. Input data management tools, and output analysis and presentation tools. Without such tools, a model is generally too complex for practical use.

What Is A User?

As Indicated above, a model and an application put some requirements on the user and vice versa. This means that only users with the right background will get the optimal benefit of a given application/model. A common misunderstanding is that you can make any model available to any user by ‘hiding’ all the expertise in automated processing and pre-coded and configured software. A more realistic scenario is probably exactly the opposite – that the more black boxes an application contains, the more it requires from the user to use and trust the application.

A user typically expects the application to do as much as possible of his repetitive work, which generally translates into a high degree of automation from input acquisition and pre-processing to preparation and evaluation of output. But automation has a price – loss of the direct feeling of what is going on. The user is easily left in a situation where he cannot trust the output simply because he does not understand the data processing behind. Consequently, the user easily either discard the application or – in worst case – makes wrong decisions based on the application results.

To ensure that the software and the procedures for use of the software are properly designed, it is a good idea to classify the users in relation to each model used in an application. If the application is optimal compressor selection based on the requirement for compression combined with physical properties as well as the cost of starting, stopping and running the individual units, the primary user will typically be a gas controller or his supervisor, but the application will depend heavily on input from other users

- unit operating costs from an accountant
- flow forecasts from a scheduler,
- maintenance plans from a facility planner,
- current state from SCADA (or a real time model)

The gas controller may be very familiar with the operation of the compressor station but inexperienced in terms of operating costs and the mathematical process of optimization. In relation to the application he may therefore be experienced when it comes to evaluating the suggested compressor configuration while he has very limited understanding of the operating cost and why the optimizer selected the specific configuration. The gas controller therefore needs a *super user* to explain/verify the operating costs and another to verify the optimization unless he has a generally positive experience with the optimizer application, i.e. many examples of acceptable results.

If the relevant super users are not available to configure and provide input for the application, the gas controller may easily reject the application and keep selecting compressor configuration as experience tells him to do.

Another example is a *scheduler* using one model (SCADA or real time hydraulic) to provide the current line fill and a scheduling model (a combination of a business and hydraulic model). The *scheduler* has no direct feeling for line pressures, available compression etc but he knows what is expected in terms of quantities and products nominated to be moved to the different destinations. Although this user understands all contractual obligations and has some feeling for the transportation capacity, he is very dependent on

- the control center to verify that the application generating the line fill is in good working order,
- the contract department to ensure that the contractual commitments are up-to-date, and

- the facility planners to provide flow capacities in accordance with available facilities.

In many cases the interaction between the different members of staff is handled manually – a time consuming exercise with a fairly long turn-around time and a high risk of misunderstandings or misinformation.

What Is IT Infrastructure?

The probably most important factor in the development of model based applications over the last 10 years has been the development in IT infrastructure. Where we 10 years ago were limited to move small amounts of information around on an old horse, we now rely on high volume, high speed carriers allowing us moving data between data sources, applications and users spread all over the company / world in almost real time.

This has opened a lot of doors, but unfortunately the concept of model-based applications has not been capable of utilizing the opportunities to the fullest extend as yet.

Issues such as global standards for application and data structures are only in their infancy, communication between users, data providers and application experts is generally constrained to emails, and slow human processing, and the entire area of relations between user skills, application architectures and expert/super user support is still widely unexplored.

The goal for many companies is to move to automated, digital communication – especially when it comes to regular exchange of large amounts of information. In this context, it is important to provide tools within all applications that allows a user to analyze and validate the out put generated in an application before it is submitted for distribution to all relevant users/applications, and after it has been received before being used.

The term 'message broker' is commonly used for tools that helps different users to exchange information transparently – even if the data structures or units send and received are not identical, or one user needs data from several independent sources. To provide the full benefits, the message brokers have to be developed to include features such as data quality management, common data standards, and handling of consequences of break downs of data sources.

How Do We Improve/Extend the Use of Model-based Applications?

The key issues are

1. to invest more time up front in understanding the needs of the company and the staff to accomplish their goals. This exercise is a joint effort between users, application experts and model experts.
2. to focus more on the applications and less on the models in the specification phase '(the model requirements will 'automatically' develop from the application specification and we – the modeling experts - know what the models can do)
3. to ensure that any application 'talks the language' of its user and that users and applications are synchronized in terms of skills and functionality.
4. to establish an open industry specific layer on top of the generic IT infrastructure offered to many different industries which provides easy exchange of pipeline relevant information.
5. to start with a fairly simple application in each relevant department but avoid involving more than two or three departments in each application
6. to establish work procedures involving the new applications before the applications are completed and user training starts (user training shall not be based on a software user manual but on company and work specific procedures – possibly supported by software manuals)

7. to establish an efficient super-user organization responsible for the applications and acting as in-house support (application support is more than and different from IT support!).
8. to design each application to provide benefits to the user in reasonable balance with the requirements to the user to provide input for other applications.

This task can be approached in many different ways. One commonly accepted base is to establish a platform of common data structures compatible with the requirements and possibilities in commonly used computer, data repository and communication tools used in other and larger industries. Whether you like it or not, the platform provided by Microsoft appears to be a good, reliable and solid platform for development in the pipeline industry.

PSIG has with the Data Standard Committee taken the first and very important step to standardize this common platform.

The next logical step is to characterize and categorize the possible applications of models from design to planning, operation and accounting. From the birth of hydraulic models, the industry has developed terms like engineering, real time and predictive model, and applications such as tracking, leak detection, look-ahead, survival time, and what-if which unfortunately only to a very limited extend has a commonly accepted and understood meaning – especially when it comes to detailed requirements and functionality. This exercise serves to make the communication between users, experts and developers more concrete, but it will also allow a more systematic analysis of the requirements to users, experts and developers to maximize the benefits of the applications.

The last part of this paper is used to illustrate some of the challenges for our industry if we shall succeed in the process of increasing the use of models in the pipeline industry, especially by making the models available to a larger audience from marketers customers and schedulers to pipeline controllers and accountants.

Firstly, we have to appreciate a few fundamental facts

1. that a model requires many types of input – some fairly static, others varying continuously.
2. that the model input shall be provided from many sources – manual as well as automatic.
3. that the individual user typically only requires a very limited fraction of the output available for his daily activities, and potentially significantly more to build confidence in the model.
4. that human nature is so that you put more pride and effort in providing accurate and timely data into a system if the system output is essential in your own work and decision process.
5. that data acquired and not used efficiently basically represent waste of resources.

Basic Hydraulic Model Applications

A hydraulic model can basically be used for

1. *Dimensioning of pipelines* based on required transportation capacities and safety limits.
2. *Validation of related data* through the more or less complex relations embedded in a model. Leak detection and automated instrument supervision are good examples of this category.
3. *Physically consistent prediction of specified data* based on forecasts of other data.
4. *Mathematical representation* of the physical pipeline in commercial and financial applications.

Application Development Process

The steps of the application development process has not changed fundamentally with the IT development, but the focus within each step has changed a lot as discussed below.

Functional Requirement Specification

The application development process always starts with identification of a need for information for decision support and/or documentation. Although ideas may (and hopefully do) develop in the process, it is essential that the application has a well-defined purpose and position in the organization (work process and user) .

From being an activity focusing on the functionality of the software, it is important to let the functional requirements focus on how the software fits into the organization together with the users and the tasks to be completed every day.

This is important in order to identify the necessary changes to the organization, education of users, and adjustment of work procedures, which basically shall be in place when the software is installed. It is important to the application users and developers to understand the environment in which the software shall operate. This process will also identify the required form, accuracy, control and timing of output.

In the cause of preparing the functional requirements it should be clear that introduction of the new software actually reduces time spend to acquire the information, improves the accuracy or timing, or reduces the work load (or any combination of these).

Model And Environment Identification

The next step is to find a way to generate the information required, i.e. to find a 'model' that can generate the information, and a processing environment that can provide the information in a timely manner.

This has traditionally been a part of the design or implementation process, but belongs more naturally in the functional requirement phase.

A mistake often seen here is just to look for the possibility of getting the output and not the process of generating it. One model may be feasible for providing the output directly, while another model may be capable of providing the output only through a manual 'trial and error' process. This may be acceptable for use once or twice a month if the model has another primary application, but definitely not for use several times a day or for use by a non-expert user.

Obviously, this activity requires a very close co-operation between the future user, the application and modeling 'experts', and maybe some prototyping to identify how much can be handled by the user and how much shall be provided by the system.

Data Source Identification

Once the model is identified, it appears to be a simple exercise to identify the necessary input but this is not always the case. Depending on the required accuracy and desired operation of the application, it is possible to generate different types of models with the same output but with different requirements for input.

As a simple example we can look at the equation above. First of all the pressure drop equation is a simplified model in itself – valid only for certain operating conditions and only with a certain accuracy. If the model shall also be applicable under non-steady conditions, the model will typically be more complex and require more input. If the accuracy requirements are limited, you may use a constant friction factor instead of a function of pipe and product properties and the flow state, i.e. an even more simple model.

No matter which model you select, you can use it to calculate flow or inlet pressure or any of the other variables but the amount of input and the solution algorithm required will depend on the model formulation selected. This is well-known to model experts and therefore the decision is often left to them, but in this process you easily forget that the input for the design decision is the users application and his requirement for accuracy. Because of the practical difficulties of predicting the accuracy of model results under actual application conditions, the proper handling of this issue is often neglected.

IT Infrastructure

Finally, it is time to figure out how to utilize a new or existing IT infrastructure to acquire input, process it, and to store, process and present output.

Efficient and reliable sharing of data – especially in connection with modeling of complex processes in and around pipelines – adds some requirements to the typical IT infrastructure whether this originates from the SCADA side (exchange of point information) or from the commercial/financial side (exchange of transactions).

Typically, a (set of) model(s) serving the main departments of a pipeline company requires data ranging from the properties of the physical assets and the products moved in the pipeline over real time process information and operating plans to nominations, contractual rules and load forecasts.

To prepare for such a model it is important not only to identify the data sources, but also to make sure,

- that the data sources are kept updated in accordance with the model requirements in terms of timing and accuracy,
- that the staff and applications to do this are prepared for the special model requirements,
- that a suitable communication infrastructure is in place for marking and identification of inaccurate or out-of-date information (status, time and responsibility tagging – is my input valid and up-to-date, and if not – who is responsible for the information?).
- that the applications and models to the greatest extend possible are designed to use and react on this communication interface. For an application utilizing all the features of the IT infrastructure including the possibility to move and store enormous amounts of data, it is impossible for the application user to manually evaluate the quality of the input.

Every user expects – not surprisingly - that his applications are working all the time. For obvious reasons, this is not possible to guarantee at an acceptable cost. If an application does not work, he expects that the application provides information on how bad it is, and in worst case what to do to get it up running again.

What is often forgotten or excluded because of costs, on complex IT systems are the tools required to help the individual users in abnormal situations. It is generally accepted that the IT infrastructure shall be monitored, and that the system manager needs a comprehensive tool set to operate and manage the networks and data bases, while the application users often are left with no or very poor tools to locate the cause of a problem and to select a corrective action.

Detailed Design And Implementation

With a *detailed* functional requirement specification, it is generally safe to leave the detailed design and implementation to IT engineers to obtain the best technical computer solution. IT systems have become so complex despite all good attempts to provide user-friendly development tools, that a reliable and cost-effective ssystem requires computer and software specialists, but even the best specification does not remove the need for close co-operation between end-users, and application, model and IT experts.

One way of reducing the need for a large group of experts on every single installation is standardization of data structures, applications and models – a development that has started but also requires a lot of openness and co-operation between pipeline companies, software developers and system integrators.

Summary

The discussion above may give the impression that model based applications requires significant resources from the pipeline company – an impression which is not very different from the experience in many pipeline companies having implemented model-based applications.

By changing the requirement, design and implementation process, it will be possible to obtain useful results faster and thereby meet two objectives

1. earlier return on the investment.
2. higher security against useless applications.

In the author's opinion, the effort and cost of implementing model-based applications will not reduce significantly in the future but the number of successful model-based applications will increase for the same investment once the right IT infrastructure and implementation process is established in the pipeline industry and the possibilities are fully utilized.

Conclusion

All this probably looks familiar to most people who in some form or another have been involved in IT projects, but despite of this we are often surprised by lack of functionality at the end of an application development. There appears to be some explanations that pops up again and again:

- lack of accurate input
- unreliable input sources, and
- incorrect assumptions made consciously or unconsciously

Further, the main emphasis is often on the 'complicated' IT and model components of the application rather than on the user and his interaction with the application. The cause of this is that the user is left out of the equation early in the process and only enters once it is too late to adjust the application.

Another frequent problem is that the pipeline company tries to justify the necessary base investment by introducing far too many applications for too many users simultaneously with the result that the first return on the investment is unnecessarily delayed.

Biography

Per Lagoni, PhD, Energy Solutions Ltd. Co-founder of Energy Solutions. Has been working in design, implementation and use of model-based applications in the pipeline industry for more than 20 years.