PROCESS MODELING

Pranshu Sharma , Satish Anand Arjunan

***Abstract*- Are you confused about the proper role of process performance models within the context of recent changes in CMMI version 1.2 high maturity guidelines? Are you struggling to understand how to use process performance models to predict progress toward achieving business goals and quality and process performance objectives? How does a project use the information gained from sub-process statistical control activities to manage process performance to achieve objectives? What is the difference between a process performance baseline and a process performance model? How are they related? If you successfully used control charts for sub-process management in the past, and were appraised at Level 4 or 5 with Version 1.1 of the CMMI, what do you have to do to be successfully re-appraised at the same level with the new model? With Version 1.2, successful quantitative project management is not just control charts! It's what you do with them and how you use them – and the answer lies within your process performance models. And oh by the way, the CMMI high maturity practices require that these models be primarily stochastic in nature. That is, each prediction should incorporate knowledge of the inherent variability of the sub-process, process, or business goal – what's that all about? This presentation addresses how mature organizations use process performance baselines and models synergistically in an integrated manner that satisfies the requirements of the CMMI. Specifically, we address how models enable us to properly use our baselines to achieve the goals and practices of quantitative management. This involves the use of multiple, linked models: • Predictions of future lower-level sub-process behavior are accomplished with statistical control models that reflect control chart behavior • Hierarchical models map these predictions up to the process level to predict progress towards reaching quality and process performance objectives •**

I. INTRODUCTION

Process models are processes of the same nature that are classified together into a model. Thus, a process model is a description of a process at the type level. Since the process model is at the type level, a process is an instantiation of it. The same process model is used repeatedly for the development of many applications and thus, has many instantiations. One possible use of a process model is to prescribe how things must/should/could be done in contrast to the process itself which is really what happens. A process model is roughly an anticipation of what the process will look like. What the process shall be will be determined during actual system development.

The goals of a process model are to be:

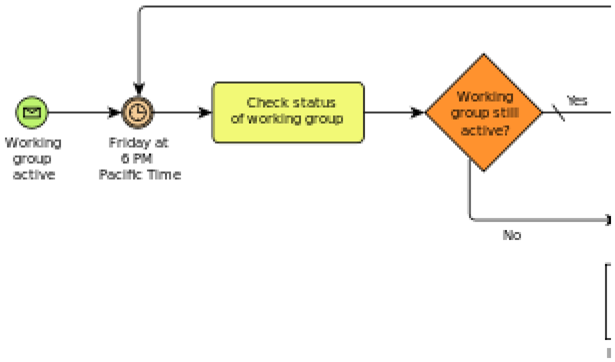
- Descriptive
    - Track what actually happens during a process
    - Take the point of view of an external observer who looks at the way a process has been performed and determines the improvements that must be made to make it perform more effectively or efficiently.
- Prescriptive
    - Define the desired processes and how they should/could/might be performed.
    - Establish rules, guidelines, and behavior patterns which, if followed, would lead to the desired process performance. They can range from strict enforcement to flexible guidance.
- Explanatory
    - Provide explanations about the rationale of processes.
    - Explore and evaluate the several possible courses of action based on rational arguments.
    - Establish an explicit link between processes and the requirements that the model needs to fulfill.
    - Pre-defines points at which data can be extracted for reporting purposes.

II. RELATED MODELS

Business process modeling (BPM) in systems engineering is the activity of representing processes of an enterprise, so that the current process may be analyzed or improved. BPM is typically performed by business analysts, who provide expertise in the modeling discipline; by subject matter experts, who have specialized knowledge of the processes being modeled; or more commonly by a team comprising both. The business objective is often to increase process speed or reduce cycle time; to increase quality; or to reduce costs, such as labor, materials, scrap, or capital costs. In practice, a management decision to invest in business process modeling is often motivated by the need to document requirements for an information technology project.

Change management programs are typically involved to put any improved business processes into practice.

With advances in software design, the vision of BPM models becoming fully executable (and capable of simulations and round-trip engineering) is coming closer to reality.



III. MODELING USING EQUATIONS

Sequential modular modeling - flowsheeting

Description

Sequential modular modeling underlies most of the flowsheet simulation programs developed since Kellogg announced their flexible flowsheeting program in 1958. The approach they and almost everyone following took was for skilled modelers to develop Fortran subroutines to model each of theEquation-based modeling 3 various types of unit operations that we use to construct complete processes.

There are subroutines for the flash unit, distillation columns, absorbers, a variety of reactor types, compressors, pumps, valves, and so forth. One constructs a complete process model by wiring up an appropriate set of these building blocks. The flowsheeting system then solves the total process model by calling each of the unit models in turn, according to how they are wired together, iterating where necessary to coverge complex process models.

The clients for sequential modular flowsheeting systems are all those people who wish to develop the heat and material balances for chemical processes.

These include process engineers, sales persons (who may have minimal technical training), PhD researchers, and the like. The systems must work and must warn of failure when they do not. Ideally they can suggest why they fail when they do in terms the user is likely to understand.

The main assumption for each unit subroutine in such systems is that its input streams are fixed and that it will compute the flows, temperature and pressure for each of the streams leaving the unit. Virtually all unit models, except for a stream mixer, require one to specify other parameters to fix their operation.

For example, a conventional flash unit requires one to specify two other things about it, such as the temperature and pressure at which it operates or the heat added/ removed and the pressure, in order for it to be a well-posed model. We call the former an isothermal flash computation while the latter is a variant of an adiabatic flash computation.

The developers of these unit models include all sorts of special tricks in them to make these computations robust. Of highest priority is that a unit model will converge when there is a solution for its underlying equations and that it fail reasonably when there is not. These tricks include developing initial guesses from which the equations typically converge. There are detectors in many of these codes to discover lack of convergence and then tests to decide what to try next to gain convergence. Each is often like a mini-expert system, containing every bit of knowledge a modeler knows and learns about such a model to make it work. Most of the code in such routines is to assure this robustness.

Examplle

The easiet way to understand this approach is to work our way through a simple example process [Westerberg, et al, p131-8, 1978]. We shall be describing a hypothetical flowsheeting system.

Fig. 1 is a simple flowsheet for the conversion of propylene to 2-hexene by the reaction

2 propylene (CH3CH=CH2) --> 2-hexene (CH2=CH(CH2)3CH3)

The propylene feed is at 150 oC and 20 bar. It contains 2 mole% propane. Our goal is to investigate the performance of this process. In particular we would like to understand if the flash unit is a good enough separation device or if we should replace it with a distillation column.Equation-based modeling

4

Our flowsheeting system allows us to build the model interactively using a computer workstation on which we can place icons for each type of unit and "wire" these icons together with streams.

We use the graphical user interface to pull icons for each type of unit from a menu onto our workspace, placing them roughly as shown above. We ask the system to draw each stream connecting two units by clicking on the output node of one unit and the input node of another. In a matter of few minutes we have the above drawing on the computer screen. From this drawing, the flowsheeting system knows the existence of all our streams and units and how they are interconnected.

Our flowsheeting system next asks us the components we wish to use in our simulation. Different streams can have different species in them so we pick a stream and pick the species for it from a list the system offers to us. If a unit insists on the same species in two or more of its associated streams, it will propagate what we pick to these streams. For example, the compressor will insist that the species that enter must be exactly those that leave. The mixer can have different species in its input streams. The species for the output stream, on the other hand, must be all the species entering.

We must also pick the physical property methods we want the system to use. In our system, these options can vary from unit to unit. For a start we pick ideal for all units. The system will use ideal mixing models for vapor and liquid mixture Gibbs free energies, enthalpies and volumes (the information it will need to compute vapor/liquid equilibrium in the flash, effect of pressure changes for the compressor, and heat balances throughout).

The system analyzes our flowsheet and tells us the flowsheet has a recycle in it. It tells us we will have to guess conditions for one recycle stream before we start our computations. It lists the candidate streams from which we select the one we wish to guess: the compressor output, the reactor feed, the flash feed,Equation-based modeling

5

the simple splitter feed or the compressor feed. We pick the compressor output which we will later be asked guess.

Next we will have to select the unit model to use for each unit shown on our flowsheet. The sequential modular flowsheeting system will have in it one or more models for each of the types of unit it supports. If there are two or more models for a type of unit, they will typically range from simple to complex in their implementation. For example, the flash unit may have three models one could use for it. The first will be a simple component splitter where the we tell the model that 98% of the propylene and propane entering are to exit in the top vapor stream and 99% of the hexene in the bottom liquid stream. The second might be a constant relative volatility flash unit. For this type of flash unit we can either specify the relative volatilities for the species or ask the system to estimate them using Raoult's Law at a given temperature and pressure. Finally there will always be a rigorous flash model which uses the physical property library to compute nonideal equilibrium K-values and nonideal mixture enthalpies.

We select a unit, say the flash. The system gives us a menu of the three models we can use for it. We pick the rigorous flash model. The system then asks that we specify two added parameters out of a list of possibilities; we choose to specify the fraction of the incoming feed that will leave in the vapor stream (e.g., 50 mole%) and the pressure.

For the reactor unit, we pick a model that allows us to specify that 80% of the propylene entering will convert as it passes through this unit.

We repeat this activity for all remaining units.

The system now asks us for the needed input to fix the computation and for it to establish starting guesses. It first asks for the feed stream specifications; we tell it the flowrate, composition, temperature, pressure and that we think it is a liquid, which the system verifies. It asks for guesses for the flowrates, composition, temperature, pressure and phase (it had better be vapor) for the compressor output. Next it goes from unit to unit to ask for the operating parameters it needs to complete each unit model specification. It also asks for guesses of some of the variables for which it believes the user guesses will aid it to converge the total flowsheet model.

The flowsheeting system now solves the model. It does this by solving the mixer first as it knows the feed stream and has a guess for the recycle stream entering it. From that computation it know the reactor feed. It solves the reactor, then the flash, the heat exchanger, the simple splitter and finally the compressor. It compares the compressor output to that guessed. If these do not agree, it reguesses the recycle and repeats the solving of the units in the sequence done above. It tells us IN LARGE PRINT it was successful in converging the flowsheet in 23 interations. We interactively go from stream to stream and unit to unit to see the values associated with each. We find that the numbers produced look plausible as we investigate them. We ask for a print out of the total flowsheet.

We are now ready to play with our model. We can choose to do many different things with it. We first change from ideal to nonideal models for evaluating physical properties, choosing the methods an associated expert systemEquation-based modeling
6
suggests to us. The expert system examines the species involved, asks us our intentions (do we want speed or do we want accuracy, for example), and suggest the options.

We resolve successfully and the numbers still look good. Now we wonder the effect of changing several of the parameters on the performance of the process. For example, we might wonder what is the impact of altering the fraction of the feed to the simple splitter which exits in the bleed stream. We set up a series of computations to be carried out one after the other where this parameter varies from a tenth of a percent to 20 percent. After running these cases, we ask the system to plot several of the variable values vs. the fraction we bleed from the process.

We decide the flash unit is not giving us a pure enough hexene product. We replace the flash unit with a small distillation column and start our simulations over again. Our first model for the column is a shortcut one capable of estimating the number of trays needed and the reflux flows needed

for ideal behavior. After we switch to a rigorous column model, we need to
play with the number of trays and feed tray location as well as the reflux ratio
to get it to perform well. We find this to be a tedious exercise. We do not
replace the bleed stream as we know separating propylene and propane is
very difficult.
We next add computations for each of the units which estimate their capital
investment and operating expenses. When we also add the value of the feed
and product streams and things like the background interest rate for our
company, we are able to compute the present worth of the process for each of
the alternative ways we choose to run it. Now we are ready to turn on an
optimizer that we find is available with the flowsheeting system.
We first ask for the plant requiring the minimum investment. After playing
for a while, we solve and find the model is trying to converge to a plant with
zero flows. That reminds us we forgot to place a production constraint on the
process which we then add.

REFRENCES

1. Colette Rolland (1993). *Modeling the Requirements Engineering Process. 3rd European-Japanese Seminar on Information Modelling and Knowledge Bases*.
2. Colette Rolland and Pernici, C. Thanos (1998). *A Comprehensive View of Process Engineering. Proceedings of the 10th International Conference CAiSE'98*. B. Lecture Notes in Computer Science 1413. Springer.
3. M. Dowson (1998). *Iteration in the Software Process, Proc 9th Int. Conf. on Software Engineering*.
4. P.H. Feiler and W.S. Humphrey. (1993). *Software Process Development and Enactment: Concepts and Definitions, Proc. 2nd Int. Conf. on "Software Process"* Sianipar, C.P.M.; Yudoko, G.; Dowaki, K.; Adhiutama, A. (2014). "Physiological Concept: Visible Modeling for Feasible Design". *Applied Mechanics and Materials* **493**: 432–437. Colette Rolland (1994). *A Multi-Model View of Process Modelling. Requirements Engineering*. Vol 4, Nr 4. Springer-Verlag.
5. C. Fernström and L. Ohlsson (1991). *Integration Needs in Process Enacted Environments, Proc. 1st Int. Conf. on the Software Process*. IEEE computer Society Press.
6. A.F. Harmsen, Sjaak Brinkkemper and J.L.H. Oei (1994). *Situational Method Engineering for information Systems Project Approaches*. North Hollan
7. Colette Rolland (1997). 'A Primer for Method Engineering. Proceedings of the INFORSID Conference.
8. BJ Hommes, V Van Reijswoud, Assessing the Quality of Business Process Modeling Techniques -Proceedings of the 33rd Hawaii International Conference on System Sciences – 2000. Bullock, L. and L.T. Biegler,
9. Levenberg,
10. Marquardt,
11. Prigogine, I. and R. Defay, Chemical Thermodynamics (trans. D.H. Everett),
12. pp187-188, Longman, London (1954).
13. Reid, R.C., J.M. Prausnitz, and T.K. Sherwood, The Properties of Gases and
14. Liquids, 3rd Ed., McGraw-Hill, New York (1977).
15. Steward, D.V., "On an Approach to Techniques for the Analysis of the Structure
16. of Large Systems of Equations," SIAM Rev., 44, 321-342 (1962).
17. Taylor, natural continuation
18. Westerberg, A.W., and S.W. Director,