ABSTRACT – We present an initiative for education in Process System Engineering (PSE) covering industrial applications in both prescriptive and predictive analytics. Prescriptive analytics or decision-automation is the science of automating the decision-making of any physical system with respect to its design, planning, scheduling, control and operation using any combination of optimization, heuristic, machine-learning and cyber-physical algorithms. Predictive analytics or data-analytics is the science of examining raw data with the purpose of drawing conclusions on the behavior of the systems using data reconciliation and parameter estimation techniques within real-time optimization and control environments. Examples for beginner, intermediate and advanced levels guide the open-users of this educational platform in PSE to evolve toward more complex problems for research, development and deployment of industrial applications in the chemical engineering and multi-related fields.

1. INTRODUCTION

The OpenIMPL initiative is a forum for the Process Systems Engineering (PSE) community exchange ideas, learnings, know-how, experiences and data using the free training license of IMPL (Industrial Modeling and Programming Language) with its underlying concepts, constructs and configurations of the Unit-Operation-Port-State Superstructure (UOPSS) and the Quantity-Logic-Quality Phenomena (QLQP). This forum is primarily intended to discuss problems found in the batch and continuous process industries when solving design, planning, scheduling, optimization, control, parameter estimation, data reconciliation and simulation examples, although interesting and suitable instances found in other industries are also welcome and encouraged. Industrial modeling frameworks (IMF) are provided as a jump-start to an implementation of industrial applications since it can be easily enhanced, extended, customized, modified, etc. to meet the diverse needs of your PSE project and as it evolves over time and use. There is no need of coding equation, but only to configure data in specific frames of the IMF’s, although the equations are constructed by modeling platform IMPL. The IMF also provide graphical user interface prototypes for drawing the flowsheet and typical Gantt charts and trend plots to view the solution of quantity, logic and quality time-profiles. Lessons learned and the importance of education in PSE and related areas as operations research can be found in Joly et al. (2015), Joly and Gut (2016) and Joly and Miyake (2017).
The data configurable in the IML file are broken-down into several categories or classes where these data categories are quantity (material flows), logic (discrete decisions) and quality (properties such as sulfur content, density, etc.). Essentially, the categories are called static (non-time-varying) and dynamic (time-varying) problem data (master and transactional) which are used to configure and control the large-scale and complex industrial optimization and estimation problems (IOP’s and IEP’s) such as planning, scheduling, control and data reconciliation and regression in either off- or on-line environments. It should also be clear from these data categories that all of these can be further classed into two higher-levels known as configuration and cycle data. Configuration data includes all data except for the cycle data found in the data categories of content (current) and command (control). Configuration data is typically static whereas cycle data is dynamic and explicitly has a time or temporal dimension attached to them to represent that the command, event, order, proviso or transaction has a defined beginning and end. The word cycle is similar to the concept of a case but hopefully provides the connotation that the IOP / IEP is executed, run or spawned on a regular / routine basis or interval most commonly referred to as the receding / moving horizon which helps to mitigate the omnipresent effects of uncertainty and variability. This is of course very well-known in the field of model predictive control (MPC) which can be likened to an on-line version off-line advanced planning and scheduling (APS) with measurement (parameter) feedback.

2. MODELING STRUCTURE IN IMPL

The Quantity-Logic-Quality Phenomena (QLQP) provides a suitable phenomenological break-down of the problem semantics and complexity where the quantity dimension details quantities such as flows, rates, holdups and yields where the quantities can be related to any stock or signal including time. The other two dimensions are logic data with setups, startups, switchovers-to-itself, shutdowns and switchover-to-others (sequence-dependent transitions) and quality data with densities, components, properties and conditions.

In addition to the QLQP, we also have what we call the Unit-Operation-Port-State Superstructure (UOPSS) that provides the arbitrary, ad hoc or anywhere-to-anywhere connectivity generally referred to as a flowsheet, topology, mapping, routings or block-diagram of the IOP / IEP in terms of the various shapes, objects or structures necessary to construct and configure it. The UOPSS is more than a single network given that it is comprised of two networks we call the physical network and the procedural network. The physical network involves the units and ports (equipment, structural) and the procedural network involves the operations (tasks) and states (activities, functional). The combination or cross-product of the two derives the projectional superstructure and it is these superstructure constructs or UOPSS keys that we apply, attach or associate specific QLQP attributes to where our projections are also known as hypothetical, logical or virtual constructs. Ultimately, when we augment the superstructure with the time or temporal dimension as well as including multiple scenarios or sites (echelons) i.e., sub-superstructures, we essentially are configuring what is known as a hyperstructure. It should be noted that in IMPL, multiple scenarios are modeled and solved simultaneously into one problem where certain variables are communed, linked or tied together to find essentially one solution to multiple sub-problems simultaneously (cf. common data) i.e., one solution to a family, group or collection of problems.

The network in Figure 1 is constructed in the UOPSS and the objects are defined as: a) unit-operations \( m \) for sources and sinks (◇), tanks (△) and continuous-processes (.fd) and b) the
connectivity involving arrows (→), inlet-ports \( i \) (⊙) and outlet-ports \( j \) (⊗). Unit-operations and arrows have binary and continuous variables (y and x, respectively) and the ports can hold the states as process yields or properties. The port-states \( j' \) and \( i'' \) represent upstream and downstream ports connected, respectively, to the in-port \( i \) and out-port \( j \) of a unit-operation \( m \).

UOPSS (Unit-Operation-Port-State Superstructure)

![UOPSS Diagram](image)

Figure 1: UOPSS framework embedded in IMPL.

3. TUTORIAL EXAMPLES

The three levels of education in PSE considers classical examples as linear (LP), nonlinear (NLP) and mixed-integer linear (MILP) programming as the following.

3.1. Beginners

The main examples in this level initiates the PSE users to the classical pooling problem from Harvelly (NLP) and the jobshop scheduling (MILP). Static and dynamic data reconciliation types of problems illustrate the use of predictive analytics.

3.2. Intermediate Users

The maritime shipping and the pipeline scheduling operations (both MILP) are representative cases of prescriptive analytics in the level. Dither signal for closed-loop estimation and hydrid dynamic simulation are found in the predictive analytics field.

3.3. Advanced Users

Advanced process control examples and more industrial example are found in this level such as crude-oil blend scheduling (MILP+NLP), retrofit planning of process networks (MILP), multi-utility scheduling operations (MILP). Advanced property tracking/tracing, time series estimation, among other predictive analytics problems are examples in this level.

4. COMPLEX PROBLEMS

The philosophy or intent of using calculation data is to minimize as much as possible the use of actual or raw numbers found in the IML file. By using names (symbols) instead of numbers (scalars) it is easier to separate the underlying model from the data. In addition, it is also convenient for the user to configure calculation expressions where any field that expects a number can be replaced by a formula. Coupled with the built-in conditional or logical functions or formulas such as IF, NOT, …, XOR, relatively complex rules can be coded or programmed to provide the user
with the necessary capability to pre- and postprogram the data. It should be noted that these functions are similar to those found in spreadsheet software such as Microsoft Excel. However, more advanced pre- and post-programming should be relegated to IMPL’s IPL (Industrial Programming Language) where its code can be embedded into any computer programming language though IPL can also configure calculations (cf. IPL’s IMPLreceiveCalc). Also note that the rules are distinctly different from constraints in IMPL; only variables and constraints are known to IMPL during the modeling and solving process. A (pre-processing) rule can only be applied to the model and cycle data BEFORE IMPL models and solves the problem and NOT during its solution. A (post-processing) rule can be applied to the solution data after IMPL has been modeled and solved where rules can be employed to alter or modify the solution data and then IMPL re-run in a loop to iteratively or sequentially arrive at good feasible solutions.

5. CONCLUSIONS

High-skilled chemical engineers are demanded in the new job positions of the companies into the Industry 4.0 environment (JOLY et al., 2017). Therefore, initiatives to improve the programming skills to able the chemical engineering students are welcomed (SANTOS, 2018). The OpenIMPL forum permits the smooth evolution from beginner to advanced levels by configuring the prescriptive and predictive analytics problems by using IMF language. More complex cases, with pre- and post-calculation to integrate to further solutions need more advanced skills in coding.

6. ACKNOWLEDGMENTS

The first author is grateful for the financial support from National Council for the Improvement of Higher Education (CAPES) in the process 2017/3300201.

7. REFERENCES

JOLY, M; GUT, J. A. W. Chemical Engineering and Operations Research: A Passing Flirtation or Marriage? Revista de Graduação USP, v. 1, p. 5-12, 2016.

JOLY, M.; MIYAKE, M. Y. Lessons Learned from Developing and Implementing Refinery Production Scheduling Technologies. Frontiers of Engineering Management, v. 4, p. 325-337, 2017.

JOLY, M.; ODLOAK, D.; MIYAKE, M.; MENEZES, B. C.; KELLY, J. D. Refinery Production Scheduling Toward Industry 4.0. Frontiers of Management Engineering, v. 37, p. 1877-1882, 2017.

JOLY, M.; ROCHA, R.; SOUZA, L. C. F; TAKAHASHI, M. T; MENDONCA, P. N.; MORAES, L. A. M.; QUELHAS, A. D. The Strategic Importance of Teaching Operations Research for Achieving High Performance in The Petroleum Refining Business. Education for Chemical Engineers, v. 10, p. 1-19, 2015.

SANTOS, M.; VIANNA JR., A. S.; LE ROUX, G. C. Programming Skills in the Industry 4.0: Are Chemical Engineering Students Able to Face New Problems? Education for Chemical Engineers, v. 22, p. 69-76, 2018.