Published in final edited form as:
*J Ind Inf Integr*. 2019 ; 13: . doi:10.1016/j.jii.2018.10.001.

# Standards-based integration of advanced process control and optimization

**Guodong Shao**[a,*], **Hasan Latif**[b], **Carla Martin-Villalba**[c], and **Peter Denno**[a]

[a]National Institute of Standards and Technology (NIST), Engineering Laboratory, National Institute of Standards and Technology, Gaithersburg, MD, USA

[b]Department of Industrial & Systems Engineering, NC State University, Raleigh, NC, USA

[c]Department of Informatica y Automatica, UNED, Madrid, Spain

## Abstract

Integration of process control with optimization is critical to Smart Manufacturing (SM). Oftentimes, however, the process control solutions from one vendor do not interoperate with the optimization solutions of another. Incompatibilities among the representation and format used by the vendors can impede interoperability. Without this interoperability, it is impossible to achieve the higher level of decision support essential to SM. We believe that an emerging standard, ISO 15746, can facilitate semantic interoperability and enable the integration of process control with optimization. This paper reports the implementation and validation of ISO 15746, Automation systems and integration - Integration of advanced process control and optimization (APC-O) capabilities for manufacturing systems. Guided by the standard, we modelled major components of a typical APC-O system using tools from different vendors, implemented the information models defined in the standard, and integrated key system functions such as process optimization, process control, and user interface. A chemical process case based on the Tennessee-Eastman problem is used to demonstrate the implementation and validation of the standard. We developed a simulation of the chemical process and integrated it with the APC-O system. We discuss the standard validation experience and the findings will be used to guide advance development of the standard.

### Keywords

Advanced Process Control; Optimization; Process Simulation; Smart Manufacturing; System Integration; Standard-based Integration

*Corresponding author. Tel.: 001 301-975-3625; fax: 001 301-258-9749; gshao@nist.gov.



No approval or endorsement of any commercial product by the National Institute of Standards and Technology is intended or implied. Certain commercial software systems are identified in this paper to facilitate understanding. Such identification does not imply that these software systems are necessarily the best available for the purpose.

# 1. Introduction

Smart Manufacturing Systems (SMS) make a range of planning and control decisions across the control hierarchy by systematically analyzing process and production data. Analysis, control, and optimization of SMS may involve data from various sources and various software tools, each formulating and solving problems independently. Due to the diversity of production contexts and the variety of methodologies being used, however, interoperation of the tools is impeded and overcoming this impediment requires manual, error-prone, and time-consuming effort. The only way to change this situation, as noted in the Smart Manufacturing Leadership Coalition (SMLC) report, is to develop and apply standardized data models. The SMLC report on implementing 21st Century Smart Manufacturing (SMLC 2011) states that “manufacturing data must be collected, stored, analyzed, and transmitted across all levels in that hierarchy. Highly efficient, standardized models are needed to manage, integrate, and use that data effectively and affordably.” Standardization is critical to interoperability of SMS including those with Advanced Process Control (APC).

APC refers to techniques and technologies implemented in applications across process industry such as oil and gas, petrochemicals, water and wastewater, chemicals, power, paper and pulps, pharmaceuticals, food and beverages, and others. These industries are characterized by continuous processes. Process control systems may be a distributed control system (DCS), a programmable logic controller (PLC), and/or a supervisory control computer. APC may reside in either the DCS or the supervisory computer to help optimize process performance and stabilize plant operations. The broad range of technologies and techniques that APC solutions possess enables manufacturers to improve production capacity, monitor process parameters, and operate with greater flexibility and safety (Howes et al. 2014).

According to the new market research report **“***Advanced Process Control Market by Revenue Source, Application, and Geography - Analysis and Forecast from 2014 to 2020*” (MarketsandMarkets 2015), the APC market is estimated to reach $1.5 billion by 2020; growing at a compound annual growth rate of 12 % from 2014 to 2020. Current vendors in the APC market include Aspen Technology Inc., ABB (ASEA Brown Boveri) Ltd., Honeywell International Inc., Schneider Electric, Siemens, General Electric Software Inc., Supcon, Rudolph Technologies, Rockwell Automation Inc., and Yokogawa Electric Corporation. Some of them are actively participating the development and implementation of the standards relevant for process industries and advanced process control.

There exist several standards relevant for process industry, for example, ISO 15519 provides rules and guidelines for representing measurement, control, and actuation in process control diagrams (ISO 2010). The *International Electrotechnical Commission* (IEC) 62424 defines procedures and specifications for the exchange of control-relevant data provided by Piping and Instrumentation Diagrams (P&ID) tools (IEC 2016). The IEC 62714 series specifies an engineering data exchange format for use in industrial automation systems (IEC 2014). ISO 15926 is a standard for data modeling and interoperability that uses several Semantic Web technologies to provide a lifecycle description of oil, gas, and chemical processes (ISO 2003). None of the above process-relevant standards address the modeling and data requirements for APC system integration. In the next section, we introduce the ISO 15746 (ISO 2015) standard, Automation systems and integration - Integration of advanced process control and optimization (APC-O) capabilities for manufacturing systems, as well as two related standards used in our implementation.

The main contributions of this paper are (1) a pilot implementation of integrating major modules of a typical APC-O system defined by the ISO 15746 standard; (2) Extensible Markup Language (XML) schema development for the information model in the ISO 15746 standard; (3) an approach of validation and testing of the standard for APC-O applications; and (4) formulation and solving of an optimization problem, control, and simulation of the Tennessee-Eastman (TE) chemical process. The pilot implementation integrates modules including a web-based Graphical User Interface (GUI), a cloud-based optimizer, an APC controller, and a simulation model that represents the TE chemical process. The standard framework, lifecycle workflow, activity models, and information models are implemented to support the formulation of the production optimization problem, the derivation of the optimal parameters, and the use of these parameters as setpoints for the controller to control the simulated plant. The OPC (Object linking and embedding for Process Control) - Unified Architecture (UA) communication protocol is used to enable the secure and reliable information exchange between the controller and the simulation. The implementation approach can be applied to any real-world process control problems.

The rest of the paper is organized as follows. Next section provides an overview of the relevant standards, i.e., International Society of Automation (ISA) 95, ISO 15746, and OPC-UA. Then the following section introduces the TE case and the implementation scenario for standard validation by integrating APC-O modules at different ISA 95 levels and the simulation of the chemical process. Followed by the implementation details of each system modules according to the scenario. Then we discuss the lessons learned. The final section presents the final discussion and conclusion.

## 2. ISO 15746 and related standards

In this Section, we introduce the standards we implemented and validated, i.e., ISO 15746 standard and OPC-UA. However, first we describe ISA 95, which is what ISO 15746 based upon.

### 2.1. ISA 95

ISA 95 standard (ISA 2014) provides a framework for exchanging manufacturing data between hierarchical levels in factories (See Figure 1). High-level functions at each level are achieved through the composition of lower level tasks. Level 4 concerns the business-related activities needed to manage a manufacturing organization. Manufacturing-related activities include establishing the basic plant schedule, determining inventory levels, and making sure that materials are delivered on time to the right place for production. This level determines what and when products are made; it operates on time frames of months, weeks, and days. Level 3 concerns the workflow needed to produce the desired end products prescribed in level 4. For each such product, this flow specifies which physical processes are used and in what order. For each of those processes, this level also specifies the associated workflow/

recipes. It typically operates on time frames of days, shifts, hours, and minutes. Level 2 determines parameter values needed to execute the prescribed workflow/recipes on the selected process. It monitors and controls that execution. It typically operates on time frames of hours, minutes, seconds, and sub-seconds. Level 1 concerns the activities involved in sensing and manipulating the physical processes. This level provides the data needed for monitoring; it typically operates on time frames of seconds and faster. Level 0 concerns the actual physical processes. The APC-O system defined in ISO 15746 is at and between level 2 and level 3 of ISA 95.

### 2.2. ISO 15746

The goal of the ISO 15746 standard is to facilitate the integration and interoperability of software tools that provide automation solutions to APC-O problems. The standard mainly focusses on the integration of APC-O capabilities at two levels of ISA 95. The APC-O system module at level 2 interacts with the Manufacturing Execution System (MES) at level 3. The module at level 2 provides production information to the MES, and in return accepts and executes the corresponding operational commands from the MES.

ISO 15746 has three parts: ISO 15746–1 (ISO 2015) - framework and functional model, ISO 15746–2 (ISO 2017) - activity models and information exchange, and ISO 17546–3 - Verification and Validation (V&V). ISO 15746–1 and ISO 15746–2 are International Standards (IS), ISO 15746–3 is still work-in-progress as a Committee Draft (CD). The following paragraphs provide detailed discussion of ISO 15746–1 and ISO 15746–2.

ISO 15746–1 defines a reference interoperability framework based on the ISA 95 hierarchy and specifies concepts, terms, definitions, and the associated rules for describing the required functional capabilities of APC-O systems. It is intended to help reduce the cost and risk associated with developing and implementing integrated APC-O solutions. Figure 2 shows the functional architecture of a typical APC-O system, which includes the following functional modules: soft sensor, advanced process control, optimization, and performance assessment (ISO 2015). An actual APC-O system may have more optimization and APC modules, and any number of soft sensor and performance assessment modules.

The soft sensor serves the same function as a physical sensor, except that values from soft sensors are obtained from a mathematical model of the physical sensor using experimental data and collected real-world data. Its outputs serve as inputs of the APC module and optimization module. Soft sensor techniques include first principle techniques and data-driven techniques. First principle techniques estimate variables based on the principles of chemical reaction kinetics, material balance, energy balance, and other known concepts. Data-driven techniques are used when a first principle model is not available or not accurate enough. Use of soft sensors enables increase of sampling frequency and improved accuracy.

The APC module utilizes the techniques and methodologies implemented within control systems including predictive and adaptive control strategies, e.g., model predictive control (MPC). The APC module receives input from soft sensor module, the optimization module, and the performance assessment module. Its outputs serve as inputs of the control system at level 2 and modules in the APC-O system other than the soft sensor module.

The optimization module adjusts process parameters to meet constraints and optimize performance relative to key performance indicators. The optimization is based on a mathematical model including a first principle model and/or a data driven model. The most common optimization goals are minimizing cost, maximizing throughput and/or efficiency or weighted trade-offs of these goals. The optimization module collects input data from APC-O system modules, and its outputs serve as inputs to modules in APC-O system other than soft sensor module.

The performance assessment module uses techniques and methodologies to help maintain efficient operating performance of the APC-O systems. It detects and diagnoses the system performance degradation and help determine whether the specified control/optimization performance targets and response characteristics are being met. It aids operators to analyze the operational state of the control systems and determines when a system maintenance is needed. Examples of performance assessment activities include determining the KPIs of the APC-O system, performing process data statistics for key process measurements, and tracking model biases.

ISO 15746–2 defines an information model of APC-O to facilitate integration of APC-O modules and the plant simulation. ISO 15746–2 builds on the framework described in ISO 15746–1 by defining activity models for APC-O systems and object models for data exchanges to support those activity models. Figure 3 shows an example of the information models in the standard, the top-level structure of a APC-O system information model. The modeling notation for ISO 15746–2 activity and information models is the Object Process Methodology (OPM), which formally defines the function, structure, and behavior of a APC-O system. Specified as ISO/PAS (public available specification) 19450, OPM is a conceptual modeling language and methodology for capturing knowledge and designing systems based on a minimal universal ontology of objects and processes (Dori 2016). OPCAT is a software environment for OPM (Dori 2016).

We implement the information model defined in the standard to enable the semantic interoperability. To provide an example of the information model, Figure 3 shows the top level of the information model, which references all the APC-O Modules (ISO 2017). Each of the Soft Sensor, APC, and Optimization modules has a Definition Type, which is an object type with subtypes to define the specific instantiation of the APC-O module. Examples of APCDefinition Type are MatrixMPC, ExpertSystem, and TransitionProcedure. Examples of SoftSensorDefinition Type are Equation and NeuralNetwork. Examples of OptimizationDefinition Type are SteadyStateOpt, DynamicOpt, and ExpertSystemOpt. A PerformanceAssessment module does not have a Definition Type but has KPISets, which are sets of KPIs used to evaluate performance of the APC-O system. Each APC-O Module is identified by Name and Type and may also have one or more vendor-specific attributes. Each module contains an Event Set, which is a group of events the module monitors or generates. The exact events depend not only on the type of APC-O Module but also on the type of manufacturing process the module is applied to. Events in an Event Set are objects of type APC-O Event Type, e.g., Communication Timeout, Process on Product, and Product Grade Change. Events may trigger action, i.e., a Product Grade Change event could trigger loading product-specific targets and limits into an APC module.

APC-OEvent Type is an object type defining common attributes of events used in APC-O.

Attributes of APC-OEvent Type are:

- Source - A reference to the object that generated the event
- Time - The time the event occurred
- Type - The type of event
- EventCategory - The defined grouping of events, such as Process Events or System Events
- Severity - The urgency of the event
- Vendor-Specific Attributes - Additional attributes defined by the specific APC-O package.

Every module also has a variable set. Typically, events influence the values of variables. It defines the variable types for each module (ISO 2017).

In this paper, focusing on Part 2, we develop a pilot implementation of an industrial control system and assess the effectiveness of the standard in communicating information across the control hierarchy.

### 2.3. OPC-UA

Another standard involved in our implementation is OPC-UA, which is used as the communication protocol to integrate the controller and the simulation of the chemical plant.

OPC is a communication standard for automation (Rohjans et al. 2013). OPC defines a standard set of objects, interfaces, and methods to facilitate interoperability between control devices from different manufacturers. Different automation levels connect through communication layers by means of physical media and protocols (Wagner 2003). A standardized interface is provided by OPC for exchanging process data through a client-server model. Based on this model, OPC enables vertical integration between system components. The OPC server translates information from device specific forms to a form consistent with the OPC-UA information metamodel for clients to use (Fadaei and salahshoor 2008).

There are several mature OPC protocols including DA (Data Access), AE (Alarm & Events), HDA (Historical Data Access), DX (Data Exchange), etc. Each of these protocols has unique read and write command structure that only impacts one protocol. OPC-UA (Unified Architecture) is the most recent communication protocol. It specifies the structure of sematic information models and defines how information between communication partners is transferred. It has been implemented on most commonly used platforms [OPC foundation 2017, OPC 2017). OPC-UA has been suggested as the basis for communication in Internet of Things (IoT). The objective of the OPC-UA is to cover all the requirements for platform independent system interfaces with versatile modeling capabilities that enable the communication of complex systems. Independence of platform and scalability are necessary to facilitate the integration of OPC interfaces directly into the system that runs on various platforms. In addition, access control and security are crucial requirements to ensure reliable and secure communication. OPC-UA provides the client with access to fine-grain data without the need to understand the entire complex system model. The architecture of a UA application, independent of whether it is the server or client, is structured into levels. In OPC-UA, every entity is a node. To uniquely identify a Node, each node has a NodeId, which is composed of three elements: (1) NamespaceIndex: The index an OPC-UA server uses for a namespace Uniform Resource Identifier (URI). The namespace URI identifies the naming authority defining the identifiers of NodeIds. They are stored in a namespace array; (2) IdentifierType: The format and data type of the identifier. It can be a numeric value, a string, a globally unique identifier (GUID), or an opaque value (a namespace specific format in a ByteString). System-wide and globally unique identifiers allow clients to track Nodes; (3) Identifier: The identifier for a node in the address space of an OPC-UA server.

# 3. Scenarios of implementing standards for process optimization, control, and simulation

This section introduces the TE case and presents a scenario that enables the APC-O system implementation and ISO 15746 standard validation for the TE process. The TE process is adopted from (Downs and Vogel 1993), it is a challenge problem that has been used for different purposes for more than twenty years.

## 3.1. Tennessee-eastman chemical process

The TE process has five major unit operations: a chemical reactor, a product condenser, a vapor-liquid separator, a product stripper, and a recycle compressor (See Figure 4.) The process produces two products from four exothermic, irreversible reactions. There are five process inputs, labeled A to E in the Figure. Component B is inert; process outputs G and H are primary products, and process output F is a by-product. The gases A to E flowing out of the reactor then go through the condenser. In the condenser, coolant is mixed with cold water and flows through to condense the gas into a liquid. The remaining gases and liquids are sent to the vapor liquid separator. The gas is compressed and sent back to the reactor through the recycle valve. Some of the gas is purged before it gets to the compressor to prevent a build-up. The liquid goes into the stripper that removes some of the remaining reactants. The product components, G and H, exit at the stripper. The inert component, B, and the by-product component, F, primarily exit the system as vapours from the vapor-liquid separator. There are 41 measured values in the TE process and 12 manipulated variables.

Due to the complexity of the TE challenge problem, researchers have devised a simplified version of the problem [Ricker 1993, Cameron and Gani 2011), which has similar rigor to the original TE problem. The simplified TE model combines the reactor and the separator vessel (See Figure 5). It has two input flows (Feed 1 and Feed 2) and two output flows (Feed 3 (Purge) and Feed 4 (Product D)). Through Feed 1, A and C are two gas compounds that enter the reactor, and through Feed 2, pure A is used to control the ratio between A and C. Product D, a liquid, exits through Feed 4, while the purge vapor flows out through Feed 3. This reaction is described by Eq. (1). The simplified TE problem is a multi-input multi-output, nonlinear system. It is open-loop unstable and contains fast and slow dynamics.

Researchers have implemented the simplified TE problem for various applications. For example, Zoeller et al. (1992), apply it to a carbonylation-of-methanol process (See Eq. 2 and Eq. 3.) In our implementation, we adopt the simplified TE problem to demonstrate our standard implementation and testing. A static simulation model of the process is developed using Aspen Plus. Figure 6 shows the simulation components, inputs, and outputs.

$$\mathrm{A{+}C} \rightarrow \mathrm{D} + \mathrm{Purge} \quad (1)$$

$$\text{Acetic Acid}{+}\text{Ketene} \rightarrow \text{Propynal} + \text{Water} \quad (2)$$

### 3.2. The optimization problem

Based on the assumptions and equations defined in the simplified TE model and the static simulation model (Figure 6), an optimization problem is formulated (Eq. 3–5). Eq. (4), the objective, is the instantaneous cost of producing certain amount of product D per hour (throughput or flowrate). Eq. (4) represents the relationship between the reaction rate of the system and the product flow rate based on the time-based equations from the model. The problem is assumed to be in steady state, so Eq. (4) is derived by setting Ricker's state equations to zero [18]. The cost equation naturally favors A, so Eq. (5) ensures that an ideal ratio between A and C is maintained. Values for three exogenous parameters are required, these parameters are the desired product D throughput (flowrate) in kmol per hour, the unit cost of input component A (per kmol), and the unit cost of input component C (per kmol). The design space is characterized by six variables. The variables that are manipulated to achieve minimum cost are the valve positions (as a percentage open) of Feed 1, Feed 2, and Feed 3 as well as the total pressure of the reactor. Using these variables, the valve position of Feed 4 can also be calculated. The optimal values of these five variables, together with the optimal cost, are returned after the optimization execution. The mathematical model is described below:

Minimize

$$C = \frac{1}{F_4} \times \left[C_A\left(y_{A1}\chi_1 F_{1max} + \chi_2 F_{2max} - F_4\right) + C_C\left(y_{C1}\chi_1 F_{1max} - F_4\right)\right] \quad (3)$$

such that

$$k_0\left(\frac{P}{\chi_3 C_{v3}\sqrt{P-100}}\right)^{1.6}\left(y_{A1}\chi_1 F_{1max} + \chi_2 F_{2max} - F_4\right)^{1.2}\left(y_{C1}\chi_1 F_{1max} - F_4\right)^{0.4} - F_4 \le 0 \quad (4)$$

and

$$y_{C1}\chi_1 F_{1max} \geq 0.8\left(y_{A1}\chi_1 F_{1max} + \chi_2 F_{2max}\right), \quad (5)$$

where,

$\chi_1$ is the Feed 1 valve position (%, expressed as decimal) as a design variable.

$\chi_2$ is the Feed 2 valve position (%, expressed as decimal) as a design variable

$\chi_3$ is the Purge valve position (%, expressed as decimal) as a design variable

$\chi_4 = \frac{F_4}{C_{v4}\sqrt{P - 100}}$ is the product valve position (%, expressed as decimal) as a derived variable

$P$ is the total pressure of system (kPa) as a design variable

$F_4$ is the product flow (kmol/h) as a user specified parameter

$C_A$ is the cost of A ($/kmol) as an exogenous parameter

$C_C$ is the cost of C ($/kmol) as an exogenous parameter

$y_{A1}$ is the concentration of A in Feed 1 (%, expressed as decimal) as a model parameter

$y_{C1}$ is the concentration of C in Feed 1 (%, expressed as decimal) as a model parameter

$F_{1max}$ is the maximum flowrate of Feed 1 (kmol/h) as a model parameter

$F_{2max}$ is the maximum flowrate of Feed 2 (kmol/h) as a model parameter

$k_0$ is the constant value associated with reaction as a model parameter

Table 1 provides a summary of variables and their nominal operating conditions. The variables are sorted into three categories: (1) exogenous model parameters (input parameters from the user) that remain fixed during the optimization, (2) design variables, and (3) derived variables (output variables) as the results of optimization excution. Model parameters (constants) are given by Ricker (1993).

### 3.3. Standard validation and implementation scenario

Figure 7 depicts the standard validation and implementation scenario for modeling and integration of the APC-O system modules for TE process based on ISO 15746. This scenario can be generalized and adopted by other cases. The scenario activities involve two main parts: Standard application and system module development. The right-hand side of the Figure 7 depicts the standard application, which includes (1) developing static simulation of the TE process, (2) identifying the optimization problem for the TE process, (3) developing XML schemas for the information models defined in ISO 15746, and (4) instantiating the relevant XML schemas to represent the defined optimization problem. The left-hand side of the Figure 7 depicts the development and integration of the APC-O system modules at different ISA 95 levels and the simulation of the TE process. In our implementation scenario, we have one Optimization module and one APC module. Implementation of the system module includes development of (1) the web-based GUI, which serves as part of the ISA 95 level 3 function, for accepting inputs from users and sending commands to the optimization module, (2) an optimization module for optimization problem representation and solving, (3) an APC module to perform the advanced process control functionalities for the TE process, and (4) a simulation model to represent the actual TE process.

The rational for making the GUI web-based is that with such GUIs the users need not install specific applications. Any devices with a web browser allow the users to start the application and input the required data to initiate the optimization. For the same reason, a cloud-based MATLAB is used as the optimization solver in the optimization module, so the formulated optimization problem can be submitted to the cloud for optimization service without the need of installing MATLAB locally. The optimization module helps users formulate, represent, and solve their optimization problems, all they need to do is to provide the exogenous parameters through the GUI. Potential users of such a system can be decision makers, engineers, and/or data analysts from the plant. These users are familiar with the manufacturing process, the APC-O system, and the problem for optimization. They need not be mathematicians nor optimization experts. In general, the information they need to provide to the system may include production objectives/goals, e.g., minimizing production cost, desired production throughput, and the constraints, and other user specified parameters, e.g., unit cost of the raw materials. After being entered into the system, the information is mapped to the optimization problem and formatted according to the selected XML schemas.

The XML schemas are developed based on the information models defined in the ISO15746–2 standard (Shao et al. 2016). Figure 8 is an example of the created XML schemas, i.e., XML schemas for optimization module. These XML schemas are instantiated for specific problems, information such as equations, data, variables of the problem, and model parameters can be represented as XML instances. Figure 9 shows an example of such XML instances that specifies objective function (cost), constraints, and optimization solver. The created XML instance files, as an executable optimization program, are submitted to the optimizer, a cloud-based MATLAB in this case. The derived values (optimal values of the objective and the parameters) are used as setpoints for the advanced process controller at level 2.

Process control ensures that the system follows process rules and the optimum control setpoints for current operating conditions and constraints in real time. The operating constraints for a plant are identified as part of the process design. But during the plant operations, the optimum operating conditions can change regularly owing to product throughput, process disturbances, byproduct as wastes, and economic evaluations. Therefore, it is necessary to recalculate the optimum operating conditions periodically. In this case, a MPC is modeled to control the TE process simulation. Figure 10 illustrates the information flow from user to the controller.

# 4. System module development

The system modules include a graphical user interface, APC-O modules and the simulation of the chemical process. The APC-O modules are developed by following the APC-O system development guidance defined in the standard. Integration and communication between system modules are realized by implementation of the standard information models. Each module in the left-hand side of Figure 7 is discussed in the following subsections.

## 4.1. The graphical user interface

As shown on the top of "System Modules" in Figure 7, we have developed a web-based GUI for user to enter exogenous parameters as inputs to the optimization problem. The GUI collects the users' input and assigns the data to the optimization problem formula. The input data, together with other model parameters (constants provided in the literature in this case), are used to instantiate the XML schema. The instantiated optimization problem is then submitted to the optimization module. The optimization results are also written to a XML file according to the XML schema.

Figure 11 depicts the interaction between the user, the web client, and the web server. The process is triggered by the user' starting the application and selecting a predefined optimization option, i.e., minimizing the cost in this case. The cost optimization page prompts the users to enter/select desired parameters. While the user provides the information, the system automatically checks the validity of the input in terms of data range and format. Once all the required fields are completed (See Figure 12), the GUI prompts for consolidating the information and mapping it to the optimization format. Finally, the instantiated XML file is sent to the optimizer.

## 4.2. Optimization

The optimization problem was described in Section 3.2. The optimizer is a commerical tool, cloud-based MATLAB. In this subsection, we mainly focus on the information exchange between the GUI and the MATLAB optimizer. Information is exchanged using XML file format. The contents of the XML file include three categories: users' inputs, derived values, and constants.

1. Users' inputs (See Figure 12):
   - Product flowrate (F4) – The user specifies his or her desired production throughput in kmol/h, which affects the cost/h of the process.
   - Cost of reactant A ($C_A$) – the unit cost in kmol/L of component A
   - Cost of reactant C ($C_C$)-the unit cost in kmol/L of component C
2. Derived values (optimization results):
   - Feed 1 position ($\chi_1$) – percentage open, determines flowrate of A/C combination feed

- Feed 2 position ($\chi_2$) – percentage open, determines flowrate of pure A feed
- Feed 3 position ($\chi_3$) – percentage open, determines flowrate of purge feed
- Feed 4 position ($\chi_4$) – percentage open, determined by pressure and desired product flowrate
- Pressure (P) – pressure of system, in kPa
- Cost (C) – the optimal cost of producing the given amount of product per hour

**3.** Constants: the values of these parameters are fixed for a given model, so in this implementation they are coded into the optimization as constant values:

- Concentration of A in Feed 1 ($y_{A1}$)
- Concentration of C in Feed 1 ($y_{C1}$)
- Maximum flowrate of Feed 1 ($F_{1max}$)
- Maximum flowrate of Feed 2 ($F_{2max}$)
- Maximum flowrate of Feed 3 ($F_{3max}$)
- Maximum flowrate of Feed 4 ($F_{4max}$)
- Constant of isothermal operation ($k_0$)

A MATLAB code parser is developed to parse the XML file to find the necessary exogenous parameters, inserts them into the optimization equation, and then prints out the derived values for the user and the controller.

### 4.3. Advanced Process Control

In this implementation, the MPC control strategy is applied. A model predictive controller is part of a multi-level control hierarchy in modern processing plants (Qin and Badgwell 2003). It bridges two levels of ISA 95, i.e., level 2 and level 1. Aspen DMC3 is used to model the MPC. The logical structure of a MPC controller includes control objectives, variables, various tuning parameters, and constraints. The logical structure of a MPC controller with inputs, outputs, control, and mechanism are shown in Figure 13. In MPC design, three different types of variables are used: manipulated variables (MV), controlled variables (CV), and disturbance variables (DV). In this case, the manipulated variables are three actual valve positions: $U_1$, $U_2$, and $U_3$ respectively. The target values of three valve positions are set from output of optimization, i.e., $\chi_1$, $\chi_2$, $\chi_3$. The controlled variables are product flowrate ($F_4$), pressure (P), and the concentration of A in the purge ($Y_{A3}$). The relationship between controlled variables and manipulated variables are adapted from Ricker (1993). The connections are derived in transfer function format from a state space model of The TE Problem.

The constraints of the model are given as below.

- Pressure (P) has an upper bound (P < 3000 Kpa).
- A in the purge (YA3) has a range (0.429 < yA3 < 0.886).
- Product flow (F4) has a set point (100 Kmol/h).
- All three manipulated variables are unconstrained.

$$y = \begin{bmatrix} F_4 \\ P \\ y_{A3} \end{bmatrix} = Gu = \begin{bmatrix} g_{11} & 0 & 0 \\ g_{21} & 0 & g_{23} \\ 0 & g_{32} & 0 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \quad (7)$$

$$g_{11} = \frac{1.7}{0.75s + 1} \quad (8)$$

$$g_{21} = \frac{45(5.67s + 1)}{2.5s^2 + 10.25s + 1} \quad (9)$$

$$g_{23} = \frac{-23.81.5 - 2.086}{s^2 + 7.874s + 0.1915} \quad (10)$$

$$g_{32} = \frac{1.5}{10s + 1} e^{-0.1s} \quad (11)$$

Using the transfer functions (8) to (11), model components are created in Aspen DMC3, which allows different types of state space model to be stored in a library. The model components in the library can be reused for express the relation between manipulated and controlled variables. For instance, the first order transfer function’s library model formula is $\frac{K}{T^*s + 1} \in^{-D^*s}$, where T is a time constant D is Delay K is Gain

In the transfer function (11), T = 10 min, K = 1.5, and D = 0.1. The graph of transfer function g32 modelled in Aspen DMC3 is given in Figure 14.

### 4.4. TE Process Simulation

In this work, a simulation of the industrial process is used in lieu of an actual industrial deployment. This simulation uses Modelica language and a Modelica simulation tool (solver) (Modelica 2017). The Modelica language is a de facto standard modeling language for object-oriented description of hybrid differential-algebraic equations (DAE) model.

Modelica is developed by Modelica Association. Owing to its object-orientation, Modelica language facilitate model development and reuse. Modelica-based modeling and simulation environments automatically perform manipulations on the Modelica model to translate it into an efficiently solvable form. A list of the available Modelica environments, both commercial and open source, can be found in Modelica Assoc. (Modelica 2017). For example, Dymola (Dymola 2018) is a commercial Modelica environment and OpenModelica (OpenModelica 2017) is an open-source environment.

A Modelica library named "TESimplified" is developed to describe the dynamic behavior of the simplified TE process and it is based on the mathematical description in Ricker (1993). The left-hand side of Figure 15 shows the Modelica model library structure. TESimplified includes a model named Reactor, a connector named pCon, models for setting the boundary conditions, and two other models describing the input and output source valves. The Reactor model describes the process unit that combines the behavior of the reactor and the separator. These models have been used to compose the ReactorOpenLoop model, a model describing the behavior of the plant, whose Modelica logical diagram is shown on the right-hand side of Figure 15. The model library is written in Modelica 3.3 and has been tested using Dymola 2018 and OpenModelica 1.11.0 64 bits under windows 2010.

### 4.5. Implementation of communication protocol: OPC-UA

OPC-UA has been introduced in Section 2. In this implementation, the controller, developed using Aspen DMC3, acts as an OPC client. The plant simulation, modelled in Modelica, serves as an OPC server (See Figure 16).

As an OPC client, Aspen DMC3 first start the CIM-IO (Common Information Model-IO) interface manager (OPC 2017). CIM-IO interface is a communication interface that provides a communication standard for interfacing with various Aspen modules such as InfoPlus.21 and third-party units such as Modbus, OPC servers. Through CIM-IO interface manager, the OPC-UA interface is activated and ready to communicate with the server. To make a successful OPC-UA connection, the OPC-UA client requires connecting with the OPC-UA server via nodes. The OPC server needs to identify the nodes and read the data. Therefore, the node addresses are provided in the model by standard I/O tagging, for instance, feed 1 real time position, $U_1$ is tagged as "/Objects/1:u1."

In the simulation (server) side, both Dymola and OpenModelica enable the establishment of a 32 bits OPC-UA server that runs the executables of Modelica simulation models (Johansson 2017). The OPC-UA server is implemented in OpenModelica interface (Open62541 2017), which is an open source implementation of OPC-UA written in C. We can either execute the Modelica model of the TES plant as an OPC-UA server using OpenModelica or run the executable file from the command line using the flag– embeddedServer=opc-ua. The OPC-UA enables an embedded simulation server using the corresponding OPC-UA interface (port 4841 in this case).

After the testing and deployment of the OPC-UA node connection, the communication between the Aspen MPC controller and the simulation is setup. The messages and feedbacks between the controller and the simulation can be monitored using the Aspen web Interface module. The communication history, data exchanged, and controller information can be viewed and exchanged within this module according to the users' need.

With the GUI, optimization, control, and simulation developed, various system functionalities of an APC-O are demonstrated. Figure 17 summarizes the information flow from GUI to simulation for the simplified TE problem. All the information exchange is standard-based, e.g., ISO 15746 information model, XML, OPC-UA.

## 5. Lessons learned

During validation and implementation of the standard, we found that ISO 15746 could be improved in the following respects: (1) the information model in its current form is not ready for direct industrial implementation. In an industrial deployment, the ability to trace changes to the information model would be crucial. Ostensibly, this ability is to be provided by OPM tools, however the version of OPM tool used in the standard does not support systematic manipulation and validation of the model. The newest version of OPM tool, however, does. In fact, the newer version of the OPM tool will allow one to define data types, perform calculations, generate schemas, instantiate models, and output various formats. All of these capabilities would facilitate industrial deployment. None of these matters currently because the standard does not provide the OPM information model in executable form. Were an executable form available, a schema for our model could have been generated. As things stand, one has to interpret the OPM diagrams in the standard and manually produce his/her implementation schemas. In our case, we generated XML through this manual process. Not only is this process time-consuming, but it also inhibits sharing and systematic management of the information model and its schemas. (2) lack of step-by-step implementation guideline, the current standard does not provide detail implementation examples that guide users to perform a complete case implementation step-by-step. A complete implementation technical report as an appendix of the standard would be helpful.

There are also some challenges for implementing OPC-UA. For example, setting up the OPC server and client requires through understanding of the specifications, which is a huge effort. Each tool also has its own implementation requirement. The tools for controller and the TE plant simulation also have certain limitations for implementing OPC-UA, i.e., OpenModelica only allows the setting up of 32 bits OPC-UA servers, which require a 32 bits client (the controller) to communicate. OPC-UA also specific requirement on the Random-Access Memory (RAM) of the system and the processor clock speed. Currently, it is also not possible to measure the performance of the OPC-UA connection in terms of the level of reliability and the quality of connection.

Even though some of the APC vendors and manufacturing companies in process industry have taken the similar approach when providing APC solutions, ISO 15746, as a new ISO standard, has provided users a systematic guideline for integrating APC-O systems. This paper provides the first implementation example and a formal approach for implementation, validation, and testing of the standard. Guided by the standard, we modelled major components of a typical APC-O system using tools from different vendors, implemented the information models defined in the standard, and integrated key system functions such as process optimization, process control, and user interface. The standards can be applied to similar problems by others in process industry. The implementation approach and the scenario can be reused for any real-world process control problem. The formal procedure includes (1) implementing applicable APC-O system modules, (2) instantiating the XML schemas for the information model defined in the standard with problem-specific parameters, (3) integrating and controlling the real process or the simulation model of the process with the APC-O system modules.

## 6. Conclusion and future work

The paper describes a recommended best practice for formulating and integrating advanced process control problems using ISA 95-based ISO 15746. A chemical process case based on the TE problem has been used to demonstrate the applications of this standard. We have (1) developed major modules of a typical APC-O system defined in the ISO 15746 standard and a simulation model of the TE chemical process using various tools from different vendors, (2) developed XML schemas for the information model defined in ISO 15746 to facilitate the semantic interoperability, (3) integrated the APC-O modules and the simulation using ISO 15746 and other standards such as XML and OPC-UA, and (4) developed a scenario of validating and implementing the standard for advanced process control and optimization. The implementation validates the standard by integrating various system modules developed using different tools across levels of the ISA 95 hierarchy, these modules are a web-based GUI, a cloud-based optimization (MATLAB), an APC controller (ASPEN DMC3), and a simulation (Modelica) that represents the TE chemical process. The OPC-UA communication protocol implemented enables the secure and reliable information exchange between the controller and the simulation. The implementation approach provides manufacturers with an example of applying the standard to their problems. The validation of the standard revealed issues and problems within the current version of the standard.

Future work includes (1) providing feedback to the standard development organization and updating the standard based on lessons learned from the standard validation, (2) identifying appropriate real-world case scenarios for industrial deployment, (3) extending the GUI to include more optimization objectives, and (4) implementing more modules of APC-O systems.

## Acknowledgement

The authors would like to thank Enan Rahman, April Nellis, and Weston Smith, Summer Undergraduate Research Fellowship students for their contributions on XML schema development, case analysis, and graphical user interface. Work performed by Carla Martin-Villalba was supported by the Ministerio de Economía y Competitividad of Spain (DPI2013–42941-R grant).

## Nomenclature

| | |
|---|---|
| **AE** | Alarm and Events |
| **APC** | Advanced Process Control |
| **APC-O** | Advanced Process Control and Optimization |

| | |
|---|---|
| **CD** | Committee Draft |
| **CIM-IO** | Common Information Model – IO |
| **CV** | Controlled Variables |
| **DA** | Data Access |
| **DAE** | Differential-Algebraic Equations |
| **DCS** | Distributed Control System |
| **DV** | Disturbance Variables |
| **DX** | Data Exchange |
| **GUI** | Graphical User Interface |
| **GUID** | Globally Unique Identifier |
| **HDA** | Historical Data Access |
| **IEC** | International Electrotechnical Commission |
| **IS** | International Standards |
| **ISA** | International Society of Automation |
| **ISO** | International Organization for Standardization |
| **KPI** | Key Performance Indicator |
| **MES** | Manufacturing Execution Systems |
| **MPC** | Model Predictive Control |
| **MV** | Manipulated Variables |
| **OPC** | Object linking and embedding for Process Control |
| **OPM** | Object Process Methodology |
| **PAS** | Public Available Specification |
| **P&ID** | Piping and Instrumentation Diagrams |
| **PLC** | Programmable Logic Controller |
| **RAM** | Random-Access Memory |
| **SMLC** | Smart Manufacturing Leadership Coalition |
| **SM** | Smart Manufacturing |
| **SMS** | Smart Manufacturing Systems |
| **TE** | Tennessee-Eastman |

| | |
|---|---|
| **UA** | Unified Architecture |
| **URI** | Uniform Resource Identifier |
| **V&V** | Verification and Validation |
| **XML** | Extensible Mark-up Language |

## References

[1]. Cameron I, and Gani R (2011). Product and Process Modelling: A case study approach. Elsevier.

[2]. Dori D (2016). Model-Based Systems Engineering with OPM and SysML. Springer ISBN 978-1-4939-3294-8.

[3]. Downs James J., and Vogel Ernest F.. (1993). "A plant-wide industrial process control problem." Computers & chemical engineering 17, no. 3 (1993): 245–255.

[4]. Dymola (2018). https://www.3ds.com/products-services/catia/products/dymola/latest-release/ Accessed: August 2017.

[5]. Fadaei A, and Salahshoor K (2008). "Design, and implementation of a new PID controller for networked control systems." ISA Trans. 47, pp. 351–361. [PubMed: 18692184]

[6]. Howes S, Pore JL, Mohler I, and Bolf N (2014). "Implementing Advanced Process Control for Refineries and Chemical Plants." Fuels and lubricants : journal for tribology, lubrication, application of liquid and gaseous fuels and combustion engineering, Vol.53 No.2.

[7]. IEC. (2014). "IEC 62714 Engineering Data Exchange Format for Use in Industrial Automation Systems Engineering - Automation Markup Language," via <http://standards.globalspec.com/std/9907433/iec-62714-2>. Accssed July 2017.

[8]. IEC. (2016). "IEC 62424 Representation of Process Control Engineering – Requests in P&I Diagrams and Data Exchange Between P&ID Tools and PCE-CAE Tools," via <http://standards.globalspec.com/std/10031951/iec-62424>. Accssed July 2017.

[9]. ISA. (2014). "ISA-95," via <http://www.isa-95.com/>. Accssed July 2016.

[10]. ISO. (2003). "15926–2: Industrial Automation Systems and Integration - Integration of Life-cycle Data for Process Plants Including Oil and Gas Production Facilities - Part 2: Data Model," via <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=29557>. Accssed July 2017.

[11]. ISO. (2010). 15519–1:2010. "Specification for Diagrams for Process Industry - Part 1: General Rules," via <http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=37547>. Accssed July 2017.

[12]. ISO. (2015). "ISO 15746–1: Automation Systems and Integration – Integration of Advanced Process Control and Optimization Capabilities for Manufacturing Systems – Part 1: Framework and Functional Model," ISO/TC 184/SC5.

[13]. ISO. (2017). "ISO 15746–2: Automation Systems and Integration – Integration of Advanced Process Control and Optimization Capabilities for Manufacturing Systems – Part 2: Activity Models and Information Exchange," ISO/TC 184/SC5.

[14]. Johansson CF (2017). "OpenModelica Interactive Simulation using an OPC UA client" Diploma Thesis. Dpt. of Computer and Information Science. Linkoping University.

[15]. MarketsandMarkets. (2015). Advanced Process Control Market by Revenue Source (Software And Services), Application (Oil & Gas, Petrochemicals, Water & Wastewater, Chemicals, Power, Paper & Pulp, Pharmaceuticals, Food) and by Geography-Analysis & Forecast (2014 – 2020). Via <http://www.marketsandmarkets.com/Market-Reports/advanced-process-control-market-61285471.html>. Accessed July 2017.

[16]. Modelica Association. (2017). Via <www.modelica.org>. Accessed August 2017.

[17]. OPC Foundation (2017). https://opcfoundation.org/. Accessed: August 2017.

[18]. OPC UA Services Specification (2017). https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-3address-spacemodel. Accessed: August 2017.

[19]. Open62541. (2017). via <https://open62541.org/>. Accessed August 2017.

[20]. OpenModelica (2017). Via <https://www.openmodelica.org/>. Accessed August 2017.

[21]. Qin SJ, and Badgwell TA (2003). "A survey of industrial model predictive control technology." Control Engineering Practice, Vol. 11, No. 7, pp. 733–764. DOI: 10.1016/S0967-0661(02)00186-7.

[22]. Lawrence Ricker, N.. (1993). "Model predictive control of a continuous, nonlinear, two-phase reactor." Journal of Process Control, no. 2 (1993): 109–123. DOI: 10.1016/0959-1524(93)80006-W.

[23]. Rohjans S; Piech K; Lehnhoff S (2013). "UML-Based Modelling of OPC UA Address Spaces for Power Systems." In Proceedings of the 2013 IEEE International Workshop on Intelligent Energy Systems, IWIES, Vienna, Austria.

[24]. Shao G, Denno P, Jones A, and Lu Y (2016). "Implementing the ISO 15746 Standard for Chemical Process Optimization." In proceedings of American Society of Mechanical Engineers (ASME) 2016 11th International Manufacturing Science and Engineering Conference.

[25]. SMLC. (2011). "Implementing 21 Century Smart Manufacturing, workshop summary report," 2011. Via < https://smartmanufacturingcoalition.org/sites/default/files/implementing_21st_century_smart_manufacturing_report_2011_0.pdf>. Accessed July 2015.

[26]. Wagner T (2003). "An Agent-Oriented Approach to Industrial Automation Systems," in: Agent Technologies, Infrastructures, Tools, and Applications for E-Services, edited by Carbonell JG, Siekmann J, Kowalczyk R, Müller JP, Tianfield H, Unland R, pp. 314–328. Springer Berlin Heidelberg, Berlin.

[27]. Zoeller JR, Agreda VH, Cook SL, Latterty NL, Polichnowski SW, and Pond DM (1992). "Eastman chemical company acetic anhydride process." Catalysis Today 13, no. 1, pp. 73–91.


NIST Author Manuscript
NIST Author Manuscript

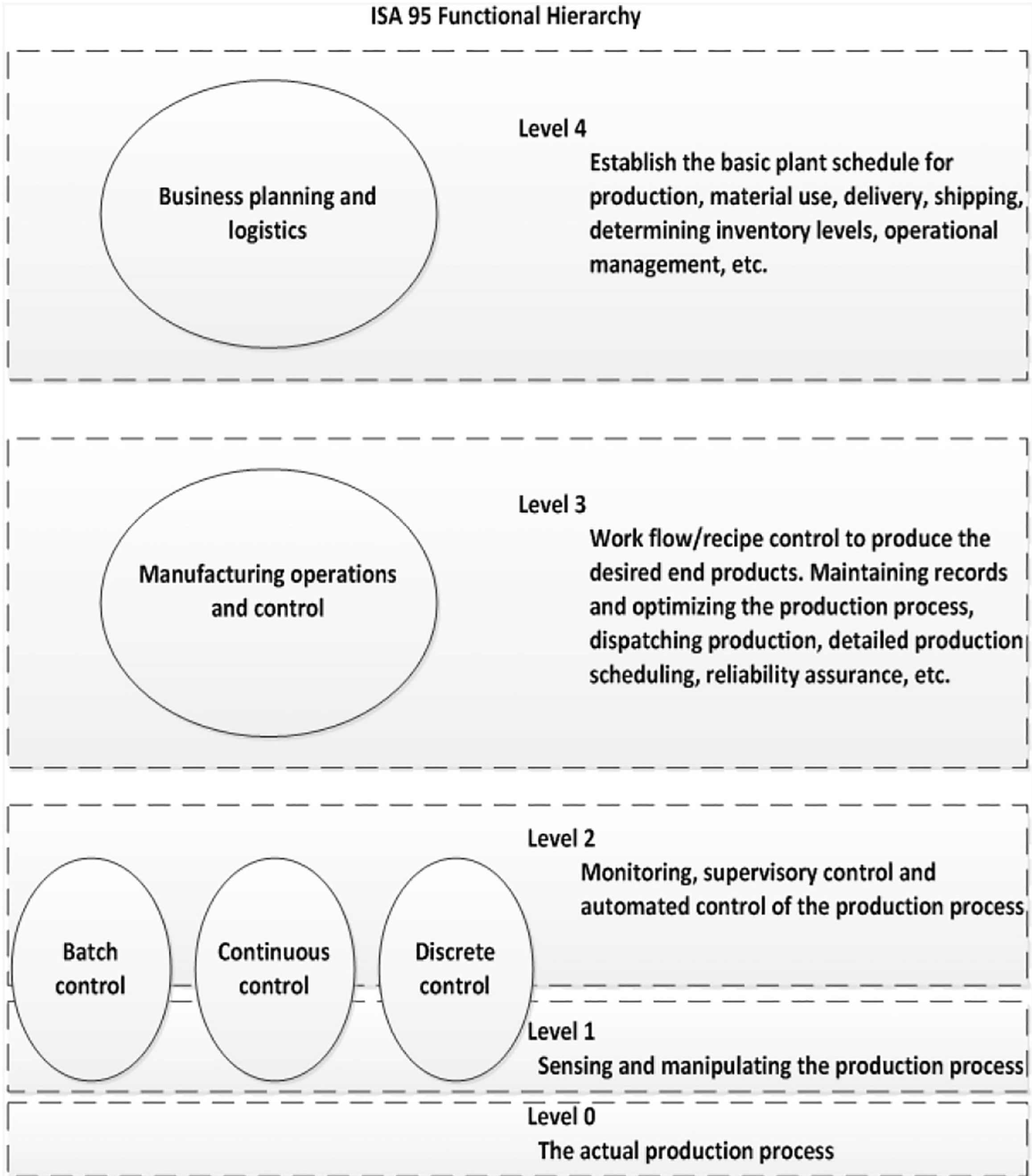


**Figure 1.**
ISA 95 levels

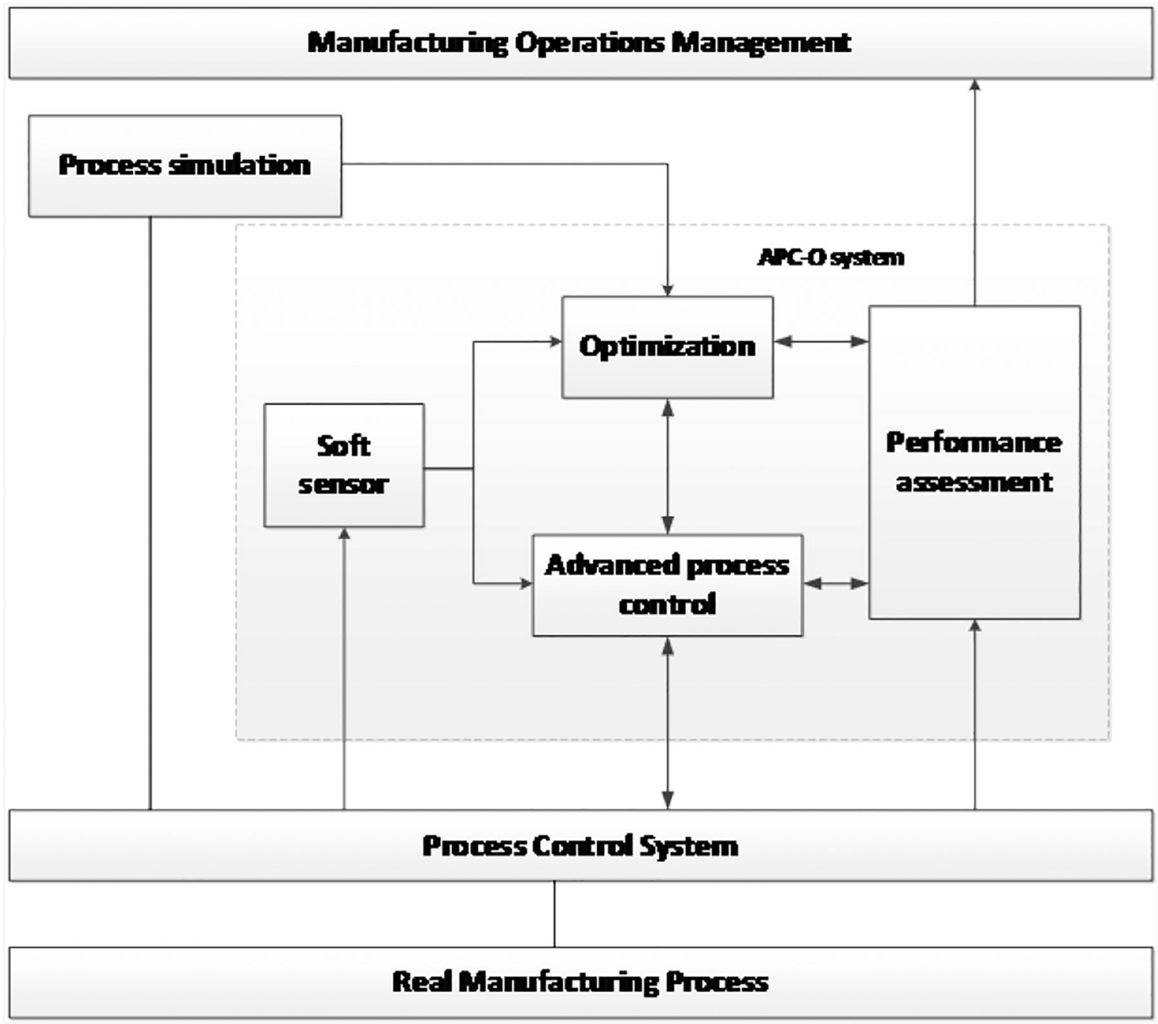


**Figure 2.**
Functional architecture of APC-O system

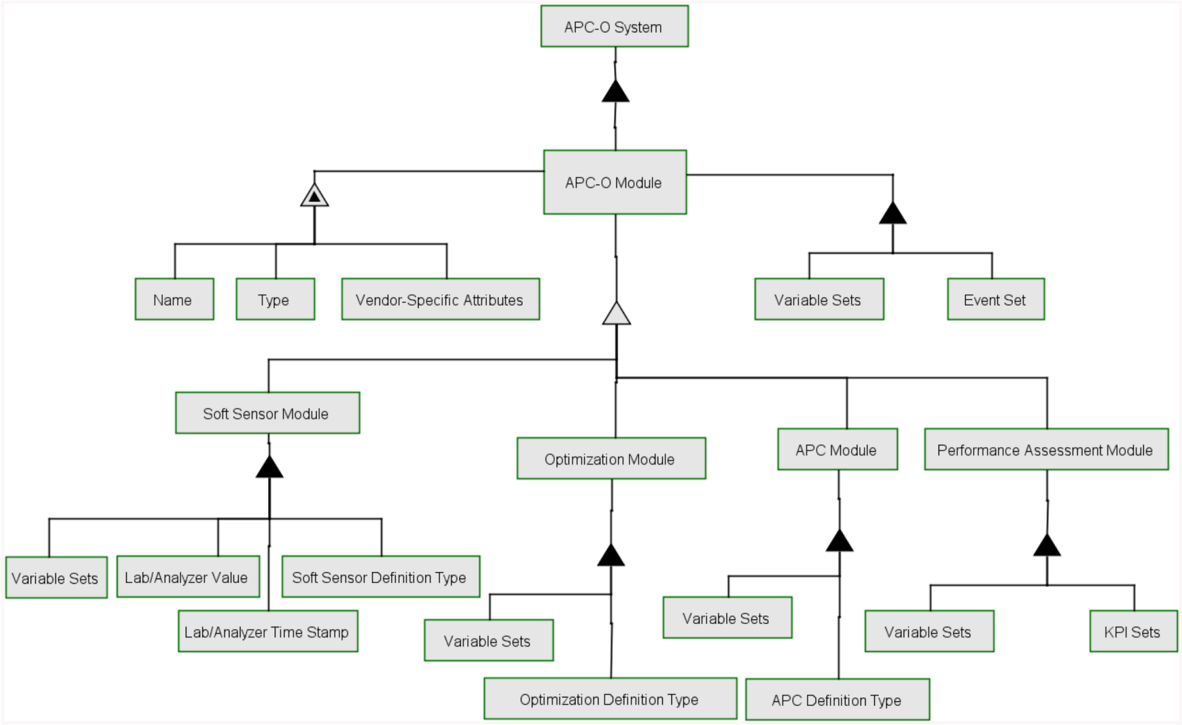


**Figure 3.**
Information model for the APC-O system

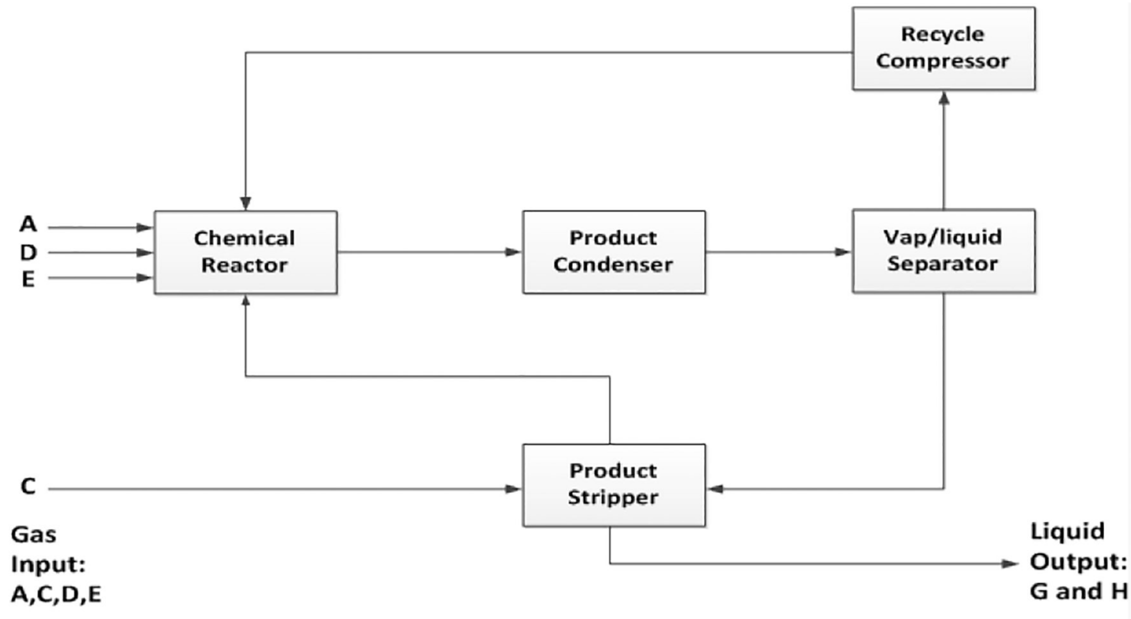


**Figure 4.**
A schematic diagram of the TE chemical process

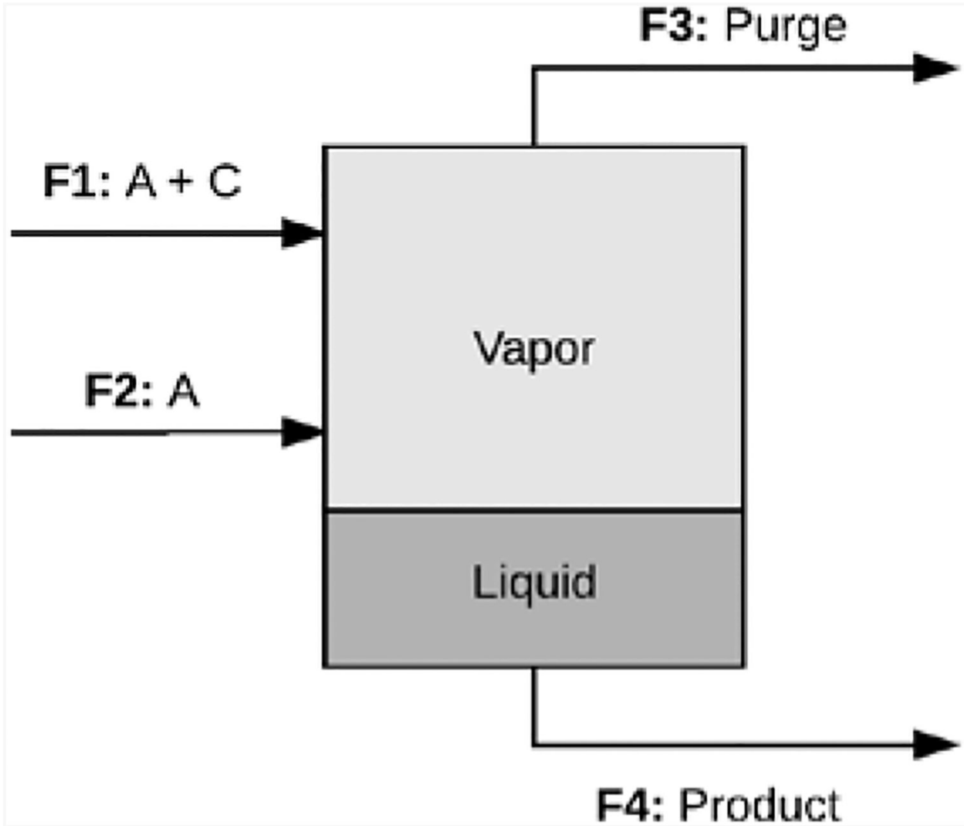


**Figure 5.**
Schematic diagram of the simplified TE process

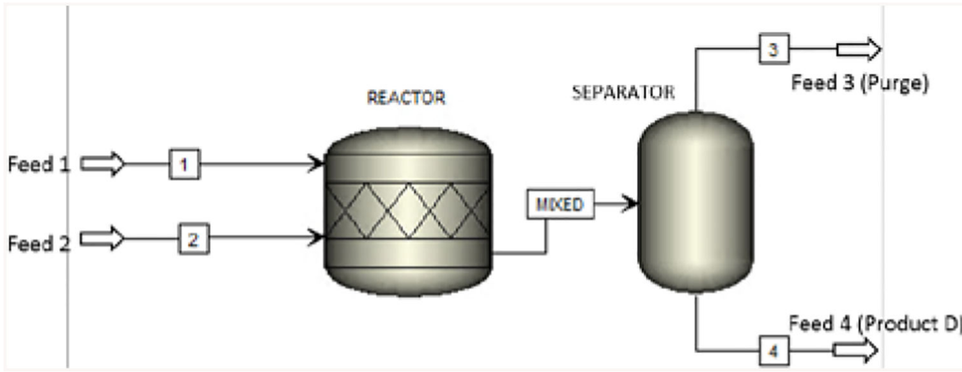


**Figure 6.**
A static simulation developed using Aspen Plus

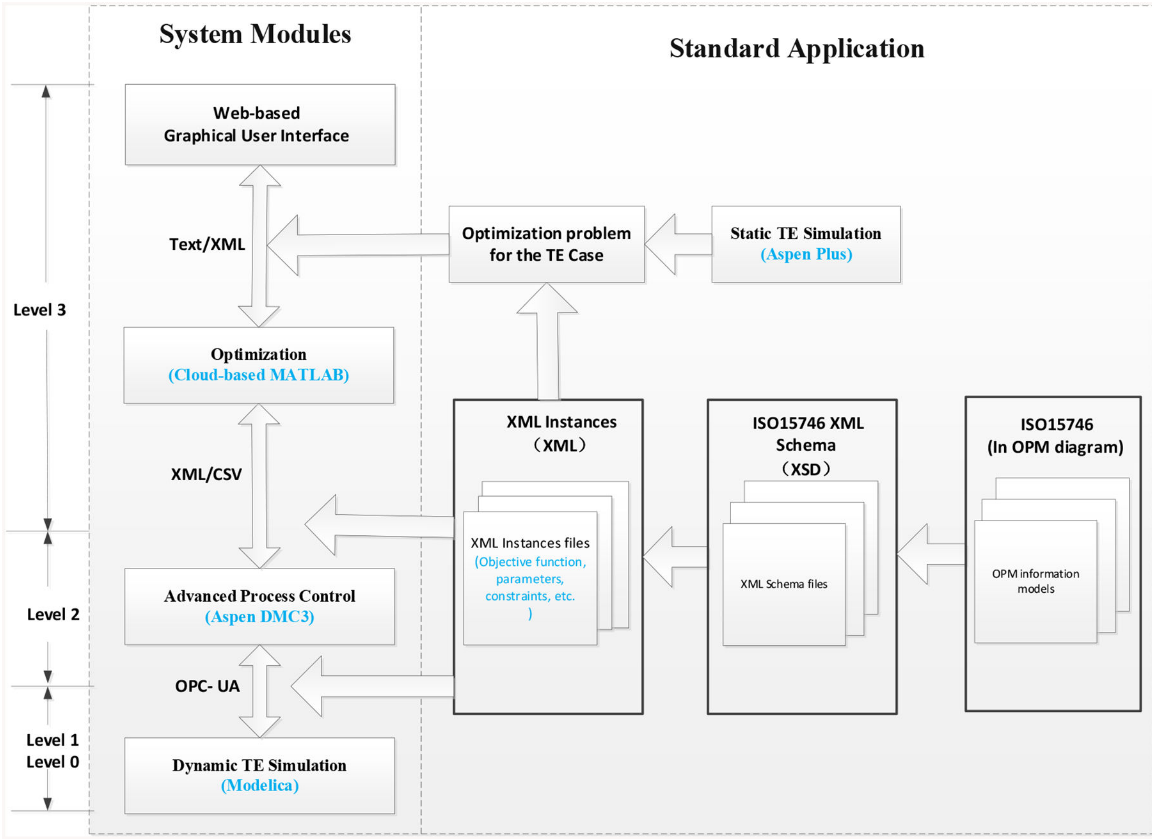


**Figure 7.**
Standard validation and implementation scenario for TE process

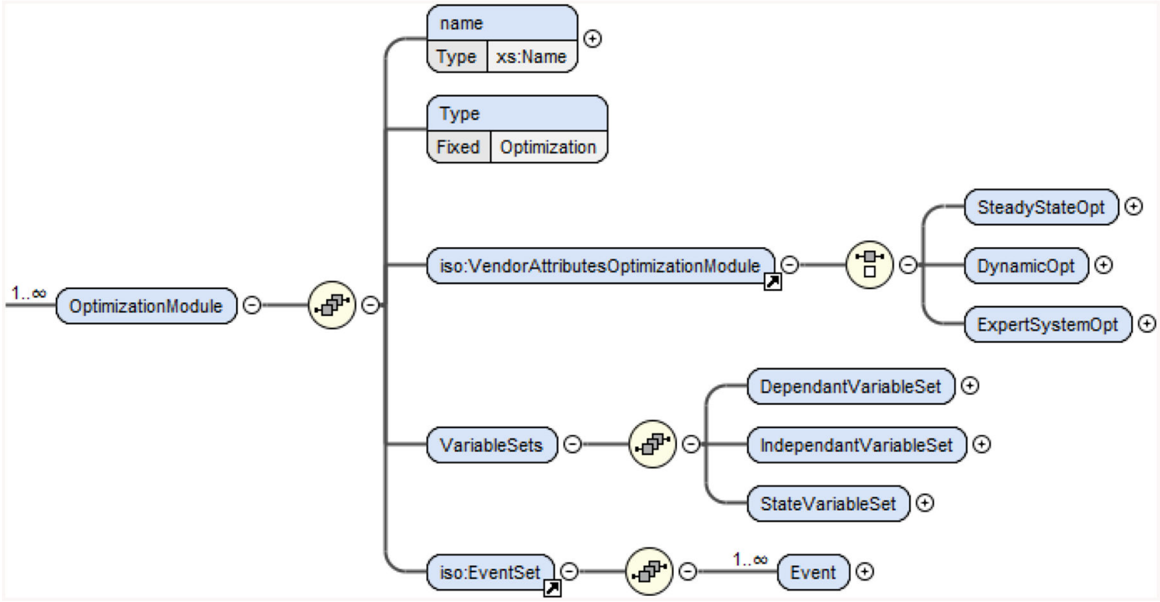


**Figure 8.**
XML schema for the optimization module

NIST Author Manuscript
NIST Author Manuscript


```xml
<iso:OptimizationModule>
    <iso:name>cost</iso:name>
    <iso:Type>Optimization</iso:Type>
    <iso:VendorAttributesOptimizationModule>
        <iso:SteadyStateOpt>
            <iso:OptimizerStatus></iso:OptimizerStatus>
            <iso:DataReconcilation></iso:DataReconcilation>
            <iso:SteadyStateSelection></iso:SteadyStateSelection>
            <iso:ObjectiveFunction>Cost</iso:ObjectiveFunction>
            <iso:ModelConstraintsSet>
                <iso:Constraint>Time &lt;= 9500</iso:Constraint>
                <iso:Constraint>Energy &lt;= 750000</iso:Constraint>
                <iso:Constraint>VolumeError &lt;= 6.5</iso:Constraint>
                <iso:Constraint>Strength &gt;= 790.5</iso:Constraint>
                <iso:Constraint>Roughness &lt;= 70</iso:Constraint>
            </iso:ModelConstraintsSet>
            <iso:Solver>CP</iso:Solver>
        </iso:SteadyStateOpt>
    </iso:VendorAttributesOptimizationModule>
    <iso:VariableSets>
        <iso:DependantVariableSet>
            <!-- Y1 -->
            <iso:Variable>
                <iso:VariableAttributeSet>
                    <iso:Name>time</iso:Name>
                    <iso:VariableID>Time</iso:VariableID>
                    <iso:DataType>float</iso:DataType>
                    <iso:ModuleType>Optimization</iso:ModuleType>
```


**Figure 9.**
An XML instance for the optimization module

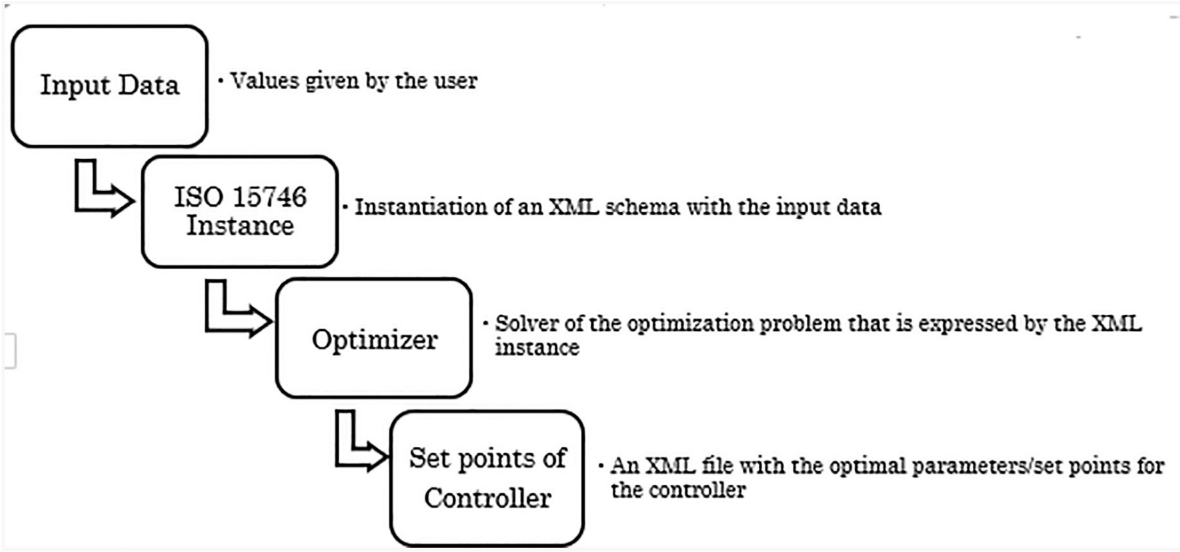


**Figure 10.**
The information flow of the GUI

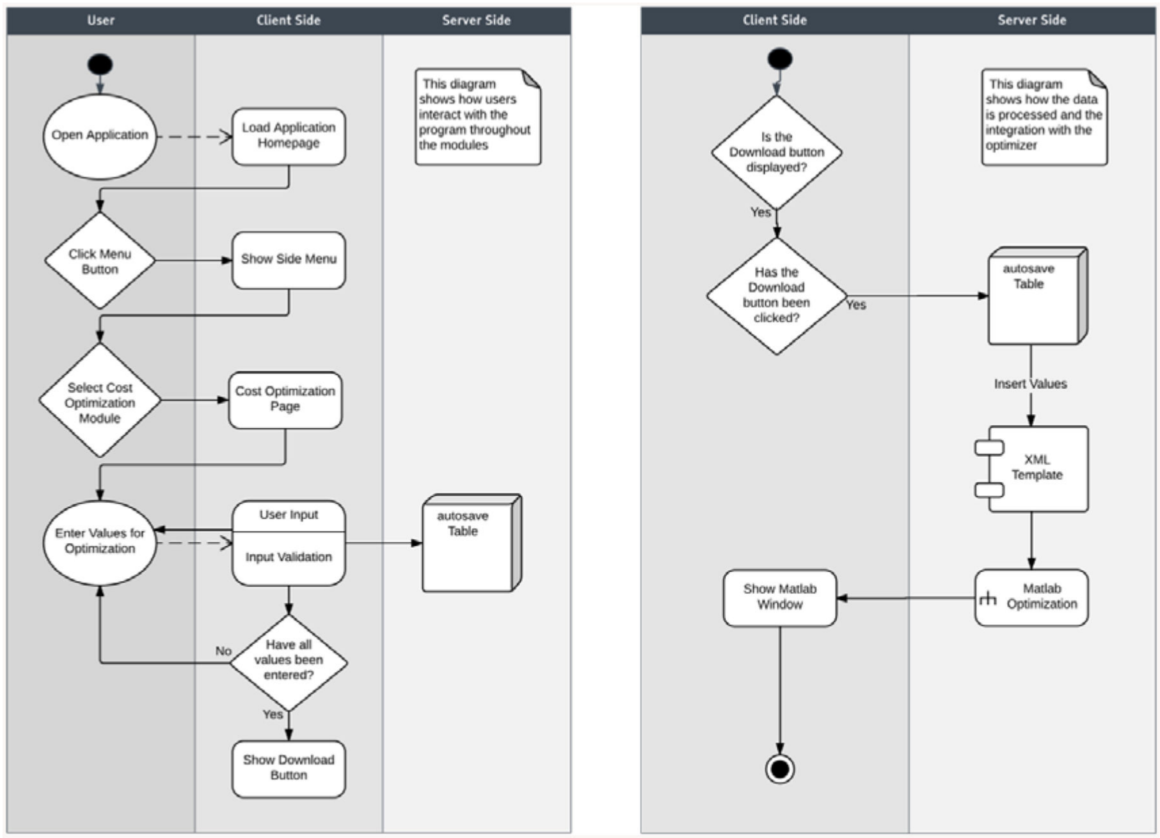


**Figure 11.**
Activity diagram of the GUI

NIST Author Manuscript

NIST Author Manuscript

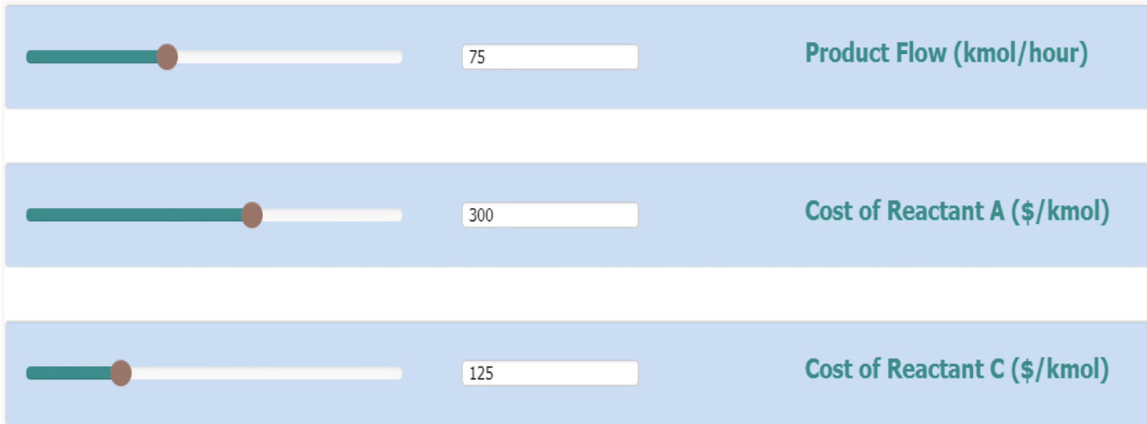


**Figure 12.**
Parameter fields of the GUI

NIST Author Manuscript
NIST Author Manuscript

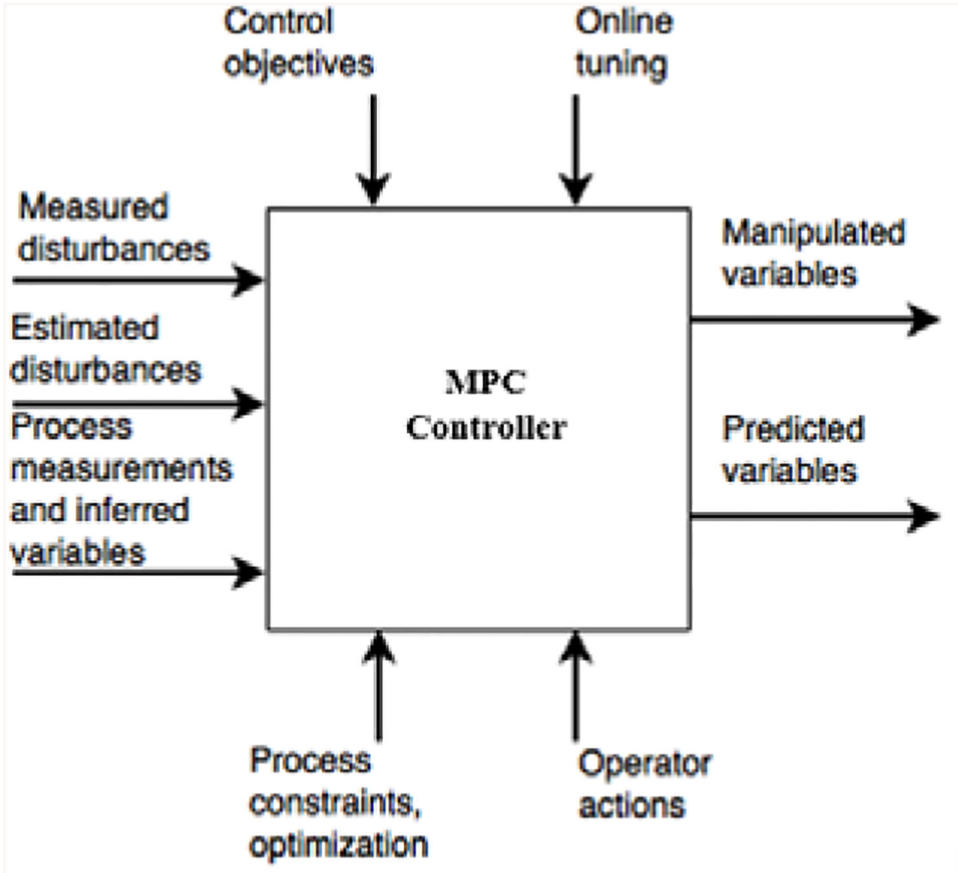


**Figure 13.**
The logical structure of a MPC controller

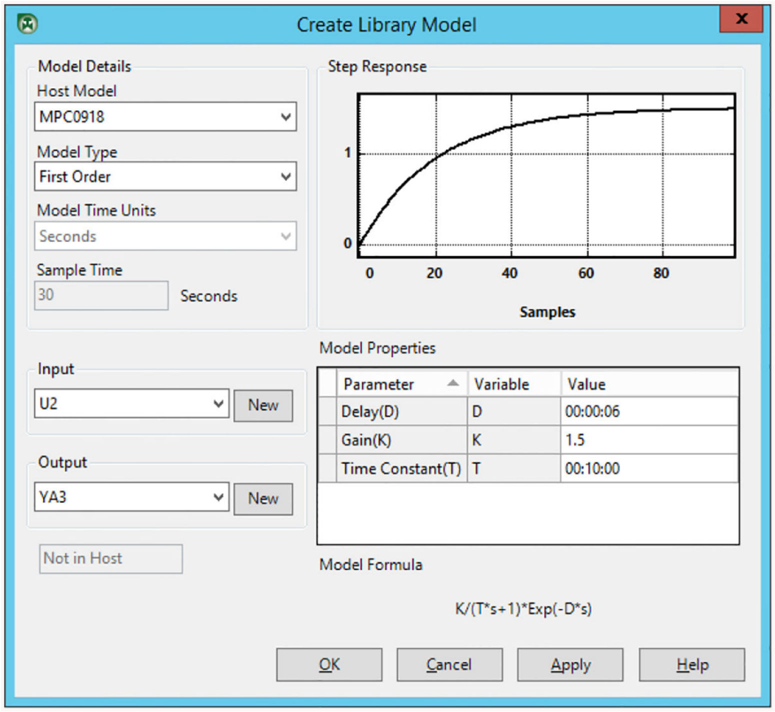


**Figure 14.**
Transfer function graph of a model predictive controller

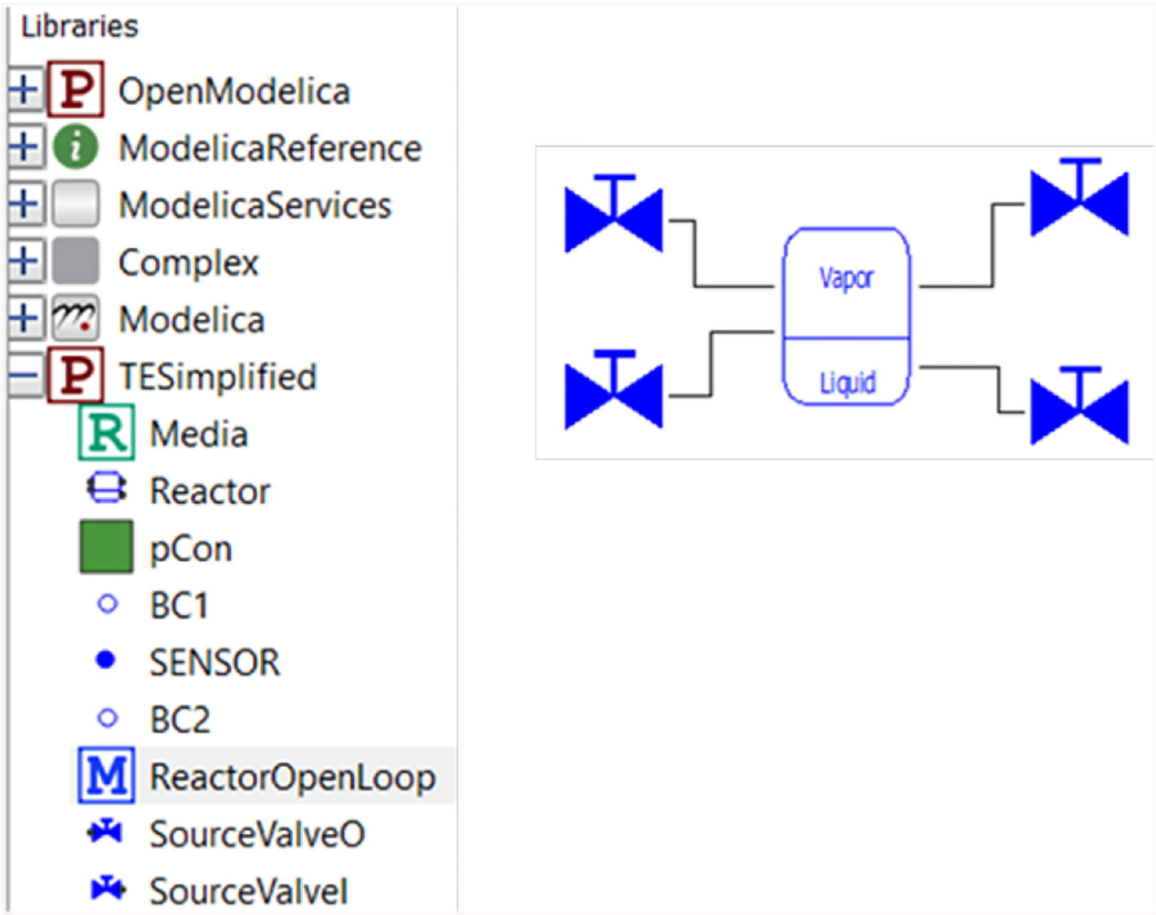


**Figure 15.**
The simplified TE Modelica model

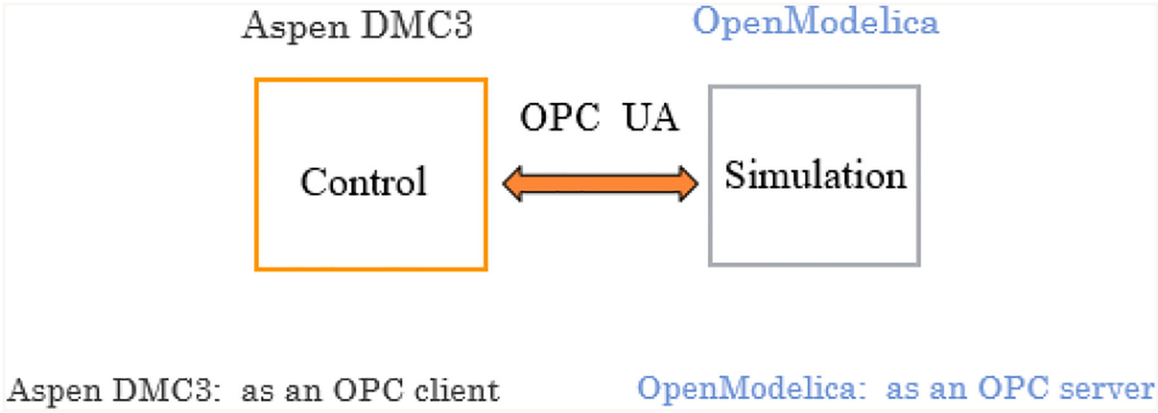


**Figure 16.**
OPC-UA architecture for Aspen DMC3 and OpenModelica

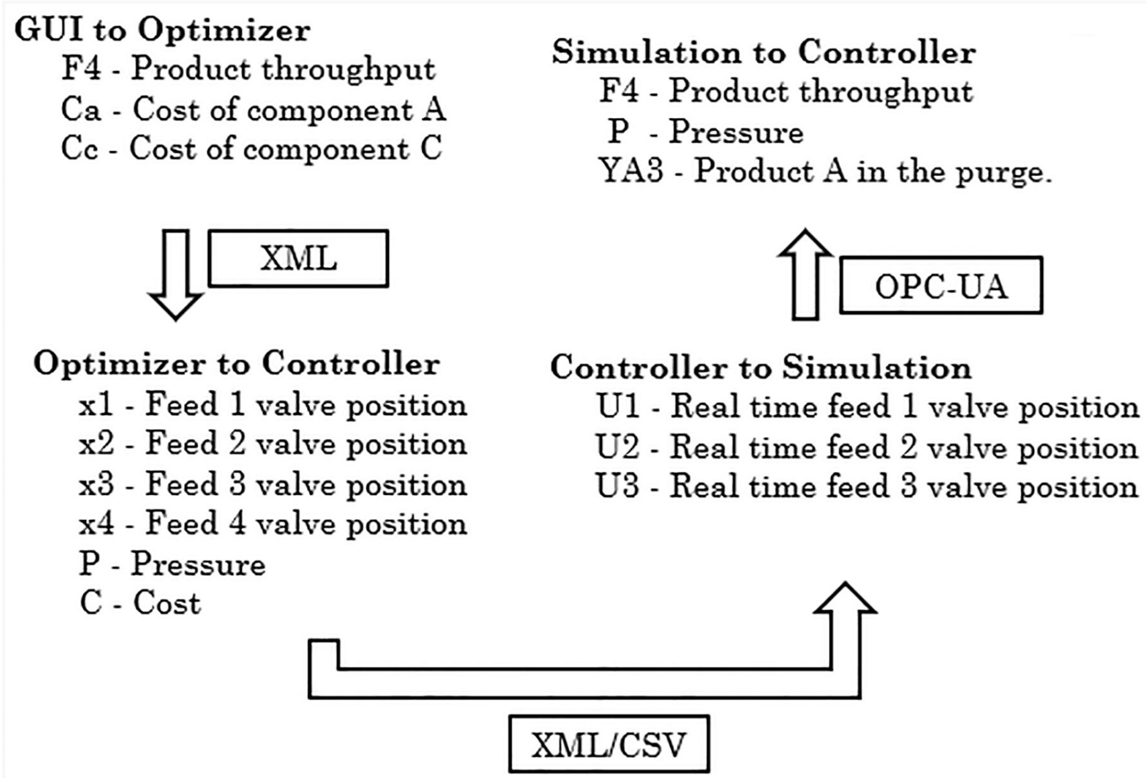


**Figure 17.**
Information flow between different module for simplified Tennessee Eastman problem

**Table 1.**

Summary of variables and nominal operating conditions

| **Output Variables** | **Set point Value** | **Description** | **Units** |
|---|---|---|---|
| $\chi_1$ | 0.610 | Feed 1 valve position | % |
| $\chi_2$ | 0.250 | Feed 2 valve position | % |
| $\chi_3$ | 0.393 | Feed 3 valve position | % |
| $\chi_4$ | 0.470 | Feed 4 valve position | % |
| $P$ | 2700 | Total system pressure | kPa |
| $C$ | 0.242 | Instantaneous cost | $/kmol |
| **Input Parameters** | **Nominal Value** | **Description** | **Units** |
| $F_4$ | 100 | Product flowrate | Kmol/hour |
| $C_a$ | 2.206 | Cost of A | $/kmol |
| $C_c$ | 6.177 | Cost of C | $/kmol |
| **Constants** | **Nominal Value** | **Description** | **Units** |
| $y_{A1}$ | 0.485 | Concentration of A in Feed 1 | % |
| $y_{C1}$ | 0.510 | Concentration of C in Feed 1 | % |
| $F_{1max}$ | 330.46 | Maximum flowrate of Feed 1 | Kmol/hour |
| $F_{2max}$ | 22.46 | Maximum flowrate of Feed 2 | Kmol/hour |
| $k_0$ | 0.001 | Constant for assumed isothermic reaction | -- |