Published in final edited form as:

Proc ASME Int Conf Manuf Sci Eng. 2016 ; 2: . doi:10.1115/MSEC2016-8635.

Implementing the ISO 15746 Standard for Chemical Process Optimization

Guodong Shao,

Systems Integration Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899 USA

Albert Jones,

Systems Integration Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899 USA

Peter Denno, and

Systems Integration Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899 USA

Yan Lu

Systems Integration Division, National Institute of Standards and Technology, Gaithersburg, Maryland 20899 USA

Abstract

This paper proposes an approach to integrating advanced process control solutions with optimization (APC-O) solutions, within any factory, to enable more efficient production processes. Currently, vendors who provide the software applications that implement control solutions are isolated and relatively independent. Each such solution is designed to implement a specific task such as control, simulation, and optimization – and only that task. It is not uncommon for vendors to use different mathematical formalisms and modeling tools that produce different data representations and formats. Moreover, instead of being modeled uniformly only once, the same knowledge is often modeled multiple times – each time using a different, specialized abstraction. As a result, it is extremely difficult to integrate optimization with advanced process control.

We believe that a recent standard, International Organization for Standardization (ISO) 15746, describes a data model that can facilitate that integration. In this paper, we demonstrate a novel method of integrating advanced process control using ISO 15746 with numerical optimization. The demonstration is based on a chemical-process-optimization problem, which resides at level 2 of the International Society of Automation (ISA) 95 architecture. The inputs to that optimization problem, which are captured in the ISO 15746 data model, come in two forms: goals from level 3 and feedback from level 1. We map these inputs, using this data model, to a population of a meta-model of the optimization problem for a chemical process. Serialization of the metamodel

DISCLAIMER

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.

This material is declared a work of the U.S. Government and is not subject to copyright protection in the United States. Approved for public release; distribution is unlimited.

population provides input to a numerical optimization code of the optimization problem. The results of this integrated process, which is automated, provide the solution to the originally selected, level 2 optimization problem.

INTRODUCTION

Smart Manufacturing Systems (SMS) make a range of planning and control decisions at all levels of the factory hierarchy. Data are critical inputs to, and outputs from, the decision-making process. In fact, according to The Smart Manufacturing Leadership Coalition (SMLC) report on implementing 21st Century Smart Manufacturing [1], large amounts of data must be collected, stored, analyzed, and transmitted across all levels in that hierarchy. The report went on to say that highly efficient, standardized models are needed to manage, integrate, and use that data effectively and affordably.

Today, making planning and control decisions may involve using software tools to formulate and solve multi-criteria optimization problems. Due to the diversity of application environments and the variety of methodologies they implement, however, these tools are often isolated and relatively independent. Because of this, each tool typically requires specialized data formats, and abstractions. This means that, instead of being modeled uniformly once, the same data is often modeled multiple times using different modeling formalisms. Developing, reusing, and integrating the models based on these different formalisms is still a manual, error-prone, and time-consuming activity. The only way to change this situation, as noted in the SMLC report, is to develop standardized data models.

In the process industries, ISO 15519 provides rules and guidelines for representing measurement, control, and actuation in process control diagrams [2]. ISO 10628 defines Piping and Instrumentation Diagrams (P&ID) [5], which capture the functional relationships among piping, instrumentation and system equipment. Several tools are available to create these diagrams. The International Electrotechnical Commission (IEC) 62424 defines procedures and specifications for the exchange of control-relevant data provided by those P&ID tools [3]. ISO 15926 is a standard for data modeling and interoperability that uses several Semantic Web technologies to provide a lifecycle description of a variety of oil, gas, and chemical processes [5].

However, none of these process-related standards addresses the data needed to integrate engineering optimization (O) tools with advanced process control (APC) tools. In this paper, we argue that there are two standards, one accepted and one quite new, that together may provide key to defining and representing that data. Those standards are ISA 95 [5], which does the defining and ISO 15746 [7], which does the representing. We will briefly describe how the ISA 95 hierarchy and its various levels are defined. We then describe how ISO 15746 uses those definitions to develop an interface data model between level 2 and level 3. Next, we describe how we used that data model to develop interfaces between the second and third levels of a process plant implementing a pedagogical chemical process, the Tennessee-Eastman process. We then formulate an optimization problem to demonstrate how the interface actually works using standard Extensible Markup Language (XML) and optimization programming language metamodel. OPLmetamodel is used for the

optimization model formulation and IBM CPLEX is used as the optimization solver to solve the problem.

ISA 95

The ISA-95 standard [5] provides a framework for exchanging manufacturing data between hierarchical levels in the factory. Figure 1 shows the high-level functions assigned to each level of that hierarchy. Level 4 defines 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. Level 4 determines what and when products are made; it operates on time frames of months, weeks, and days. Level 3 defines 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, Level 3 also specifies the associated recipes. Level 3 typically operates on time frames of days, shifts, hours, minutes, and seconds. Level 2 sets the parameters needed to execute the prescribed workflow/recipes on the selected process. It also monitors and controls that execution. Level 2 typically operates on time frames of hours, minutes, seconds, and sub-seconds. Level 1 defines the activities involved in sensing and manipulating the physical processes. Level 1 provides the data needed for monitoring; it typically operates on time frames of seconds and faster. Level 0 defines the actual physical processes.

In this paper, we explore a novel method of integrating the workflow/recipes and other data from Level 3 with the parameter setting optimization (O) and monitoring/control (APC) in Level 2. That integration is based on data models from a recent standard, ISO 15746. We map information from this data model to a population of a metamodel of optimization problems. Serialization of the metamodel population provides input to a numerical optimization code. Results of the optimization provide the parameter settings, which are the inputs needed for monitoring and control.

ISO 15746

The ISO 15746 standard is intended to facilitate the integration and interoperability of software tools that provide automation solutions to optimization and advanced process control (APC-O) problems. Currently, the standard has two parts: ISO 15746-1 [7] and ISO 15746-2 [8]. ISO 15746-1 defines a reference interoperability framework, based on the ISA 95 hierarchy. Its goal is to reduce the cost and risk associated with developing and implementing integrated APC-O tools. Its scope is limited to specifying the set of concepts, terms, definitions, and the associated rules for describing the required functional capabilities of those tools.

The major focus of this paper, however, is on ISO 15746-2, which defines an information model of APC-O to enable integration of different applications and systems. It builds on framework in ISO 15746-1 by defining activity models for APC-O systems and object models for data exchanges to support those activity models. In this paper, we focus on a few

information models defined in ISO/CD [8]. Table 1 shows the various symbols used in those models and their definitions.

Figure 2 shows overall structure of the information model starting from the top level as APC-O systems, which is comprised of one or more APC-O Modules [8]. An APC-O Module is identified by Name and Type and may also have one or more vendor-specific attributes that are provided through a discovery service interface. The unique attributes to the type goes into the Vendor-SpecificAttributes section within the module. Every module has an event set and a variable set. Events and variables both have their specific attributes. The top-level diagram defines what types of variables each module has in sets.

Figure 3 shows the information model for APC-OVariable type and the subtypes defined [8]. All APC-O variables Module types have VariableSets of base type APC-OVariable type. VariableSets are sets of variables used by the APC-O system. Defined variable sets are shown for each APC-OModule type. Variables in each of these sets are subtypes of APC-OVariable type, which is an object type defining common attributes of all variables used in APC-O.

Based on ISO 15746-2 [8], there is no generic OptimizationDefinitionType, every optimization tool will need to provide its own unique structure. Figure 4 shows an example of OptimizationDefinition types [8]. It does not represent specific technologies in their entirety but rather illustrate how an integration object might look.

The three instances shown are those that might be of interest external to the specific instantiation of the OptimizationModule.

- A SteadyStateOpt object represents a type of optimization where an objective function is minimized using steady state process models. The path that the process takes to achieve the optimum steady state conditions is not considered.
- A DynamicOpt object represents a type of optimization where an objective function is minimized over a fixed horizon using dynamic process models. Both the path and the final steady state conditions are considered in the solution.
- An ExpertSystemOpt object represents a type of optimization where optimum conditions are determined by a set of rules similar to if-then-else logic. Process models may be embedded in and used by the rules, but these models are not the fundamental basis for determining optimum conditions.

In this paper, we will exemplify the application of standard ISO 15746 using a chemical process, which is a continuous process.

THE TENNESSEE-EASTMAN (TE) CHEMICAL PROCESS

A schematic diagram of the TE chemical process is based on Downs and Vogel [9] as shown in Figure 5. 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. The process produces two products from four exothermic, irreversible reactions. There are five

process inputs labeled as A to E, with component B as an inert, process outputs G and H as the primary products, and process output F as a byproduct.

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 only measured value is the temperature of the cool water and the only manipulated variable is the cool water flowrate. The remaining gases and liquids are then sent to the vapor liquid separator.

The vapor/liquid separator separates vapor and liquid by using gravity to pull down the liquids into one stream while the gas is taken up to another stream. The measured values for this operation include the separator's pressure, temperature, and level. The only manipulated variable is the flowrate of the liquid leaving the 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 buildup. The measured values in this process include the purge stream flowrate, the recycle flowrate, and the work done by the compressor. The manipulated variables include the Purge valve and a recycle valve positions.

The liquid goes into stripper that removes some of the remaining reactants. This is done by sending steam and gas C around the liquid to strip them off the reactants. The measured values are the stripper's pressure, temperature, and level, the steam's flowrate and the underflow rate of the liquid products leaving the scope of the process. The product components, G and H, exit at the stripper base. The inert component, B, and the byproduct component, F, primarily exit the system as vapors from the vapor-liquid separator.

The primary goal of this process is to facilitate a number of reactions. In the expressions that represent the reactions, (g) stands for gas and (liq) stands for liquid. The specific reactions are

There are also two reactions that create the byproduct liquid F. These reactions are

The reaction rates of the endothermic reaction are a function of heat and reaction is modeled as an ideal, continuous, stirred-tank reactor with internal cooling to remove the heat of reaction. There are 41 measured values in the T-E process and 12 manipulated variables.

HOW TO USE ISO 15746 TO ACHIEVE INTEGRATION

Using ISO 15746 involves a two-stage process. In the development stage (see Figure 6), a conceptual data model provided by the standard is used for specifying corresponding XML schemas. Once created, the XML schemas can be reused for multiple implementations. Any APC-O application can be instantiated based on the schema and the XML data can be exchanged for various uses, including data management, simulation, advanced control and optimization.

In this case, the XML schemas will be used for representing data for the T-E chemical process. Figure 7 and Figure 8 are examples of the created XML schemas.

In the implementation stage, the T-E process is analyzed, equations and relevant data are derived from the MATLAB simulation [10] and literatures [9] [11], data and variables are represented as XML instances according to the developed XML schemas. Then, a subset of those XML instances, related to the optimization problem defined in (see (8) below), is used as input to level 2. Figure 9 is an example of the XML instances. Information from level 2 and level 3 systems is mapped to the optimization metamodel. Those instances are used to create an executable optimization program through the OPLmetamodel.

MAPPING THE DATA TO THE OPTIMIZATION METAMODEL

A *metamodel* is a model of a modeling language that provides sufficient detail about the modeling language that it may serve as a storage form for the language. For example, the Unified Modeling Language (UML) metamodel provides description of the concept of things such as classes and methods. Instances of the UML metamodel can describe classes (model content) with sufficient fidelity so that automated tools can generate Java class definitions from the class descriptions.

An optimization metamodel is a conceptual model for capturing, in abstract terms, the essential characteristics of a given optimization problem – such as an objective function and its constraints. The metamodel provides a schema of sufficient formality to enable the problem modeled to be serialized to statements in several, concrete optimization languages – such as the Optimization Programming Language (OPL) [12], A Mathematical Programming Language (AMPL) [13], and the General Algebraic Modeling System (GAMS) [14].

Currently, however, the metamodel developed at the National Institute of Standards and Technology (NIST) only supports OPL [15]. The OPL metamodel specification can be found on the NIST GitHub site [15]. Using that specification, we have developed software to (1) read OPL models and (2) write populations of the metamodel needed to produce complete OPL code for the problems. The purpose of (1) is to produce “templates” for classes of problems. Elements specific to the optimization problem at hand can be

substituted for placeholders in the template. The purpose of (2) is, of course, to produce the OPL input for an OPL-capable, numerical optimizer – one of the tools and solvers at the bottom of Figure 6.

An example of usage of the OPLmetamodel in chemical process manufacturing is depicted in Figure 10. The top portion of Figure 10 shows the inputs used in problem formulation (middle layer of Figure 10). Problem formulation concerns the development of design space constraints (typically mathematical inequalities) and objective function (typically an equation with weighted terms). There are many possible technical means of formulating the problem. For this problem, we simply process the XML-based data to expressions of the metamodel and substitute these for elements of the problem template. The completed OPLmetamodel population is then serialized to an optimization solver as depicted in the lower layer of Figure 10.

EXAMPLE OPTIMIZATION PROBLEM

With the created XML instance files for the case, various analytical and/or control tasks can be modeled and performed. In this section, we demonstrate the model and data transformation using the OPL metamodel, we take a subset of the XML file created for the TE chemical process as input to model a simplified optimization problem. OPLmetamodel is used for the optimization model formulation and IBM CPLEX is used as the optimization solver to solve the problem.

In the chemical case, there are four possible single input single output relationships from manipulated variable to measured value that govern the process: (1) Reactor Cool Water Flow $\rightarrow$ Reactor Temperature, (2) D-feed Flow $\rightarrow$ Reactor Level, (3) Separator Pot Liquid Flow $\rightarrow$ Separator Level, and (4) Product Liquid Flow $\rightarrow$ Stripper Level. We selected the first set, i.e., Reactor Cool Water $\rightarrow$ Reactor Temperature relationship as the base of our optimization problem. There is heat generated during the reaction process, so cool water is needed to flow through the reactor to regulate the temperature. Equations and data have been generated and collected by analyzing the corresponding MATLAB simulation program [10]. Particularly, the relationship between manipulated variables and the measured values at Mode 1 has been examined and plotted. The relevant simplified equations are listed below:

$$T_r = -1.85 C_r + 174.24 \quad (5)$$

$$R = 0.1 * T_r^2 \quad (6)$$

From Equation (5) and (6), we can derive

$$R = 0.1 * T_r^2 = 0.1 * (-1.85 * C_r + 174.24)^2, \quad (7)$$

where R: Reaction Rate, T_r : Reactor Temperature, C_r : Reactor Cool Water Flow

To identify the control limits of the valve position, we want to conduct an optimization on the reaction rate (7). In other words, we want to find the value of cool water flow rate that leads to a minimum reaction rate so as to find the lower bound of the valve position of the cool water to avoid that. The constraints of C_r is $13.2\% < C_r < 100\%$ (out of $227.1 \text{ m}^3 \text{ h}^{-1}$). Therefore, the actual optimization formulation is given by

$$\begin{aligned} \text{Min}_{st} \quad & R = 0.1 * (-1.85 * C_r + 174.56)^2 \\ & Tr = -1.85 * C_r + 174.56 \\ & 13.27\% < C_r < 100\% \end{aligned} \quad (8)$$

Figure 11 (at the end of the paper) is a CPLEX execution window that shows the optimization model produced from the optimization metamodel and the optimization result.

We find that the minimum reaction rate is at approximately 94.18 % of the cool water flow. The minimum reaction rate is zero, which means that the reaction stops at this point. The cool water flow valve should be controlled to avoid this cool water flow rate.

SUMMARY AND FUTURE WORK

In conclusion, we developed an approach for using ISA 95 and ISO 15746 (parts 1 and 2) as a foundation for integrating optimization with advanced process control (APC-O) using our recently developed optimization metamodel. We developed XML schemas for the information models defined in the standard, and applied them to a Chemical-processing-plant optimization problem at level 2 of the ISA 95 hierarchy as a demonstration of our approach. That demonstration was based on an OPL metamodel implementation. The OPL metamodel is translated to an OPL model, which is solved directly by a commercial solver, IBM CPLEX.

Future work includes (1) verifying the needs of standard-based modeling and analysis systems by industry and identifying appropriate real world case scenarios, (2) providing feedback to the standards organization responsible for developing ISO 15746, (3) developing metamodels and translators for other numerical solvers, and (4) developing dashboard for automating the modeling formulation and execution.

Acknowledgments

We would like to thank Enan Rahman, a Summer Undergraduate Research Fellowship student, for his help on XML schema development and chemical process analysis and Ibrahim Assouoko for his valuable ideas and discussion.

References

1. SMLC. Implementing 21 Century Smart Manufacturing, Workshop Summary Report. 2011. https://smartmanufacturingcoalition.org/sites/default/files/implementing_21st_century_smart_manufacturing_report_2011_0.pdf
2. ISO. Specification for Diagrams for Process Industry - Part 1: General Rules. 2010. 15519-1:2010 http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=37547

3. ISO. ISO 10628-1:2014 Diagrams for the Chemical and Petrochemical Industry - Part 1: Specification of Diagrams. 2014. http://www.iso.org/iso/catalogue_detail.htm?csnumber=51840
4. IEC. IEC 62714 Engineering Data Exchange Format for Use in Industrial Automation Systems Engineering - Automation Markup Language. 2014. <http://www.automationml.org/>
5. ISO. 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. 2003. http://www.iso.org/iso/iso_catalogue/catalogue_tc/catalogue_detail.htm?csnumber=29557
6. ISA. ISA-95. 2014. <http://www.isa-95.com/>
7. ISO. ISO 15746-1: Automation Systems and Integration – Integration of Advanced Process Control and Optimization Capabilities for Manufacturing Systems – Part 1: Framework and Functional Model. 2015. ISO/TC 184/SC5
8. ISO/CD. ISO TC184/SC 5/WG5 N073. Automation Systems and Integration – Integration of Advanced Process Control and Optimization Capabilities for Manufacturing Systems – Part 2: Activity Models and Information Exchange. 2015. ISO/TC 184/SC5
9. Downs JJ, Vogel EF. A Plant-wide Industrial Process Control Problem. Computers & Chemical Engineering. 1993; 17(3):245–255.
10. ASU. Tennessee Eastman Problem for MATLAB. Arizona State University; 1998. <http://cseel.asu.edu/?q=node/33>
11. Tian Z, Hoo K. Multiple Model-Based Control of the Tennessee-Eastman Process. Industrial & Engineering Chemistry Research. 2005; 44:3187–3202. <http://pubs.acs.org/doi/pdf/10.1021/ie0496939>. DOI: 10.1021/ie0496939
12. Van Hentenryck, P., Lustig, I., Michel, L., Puget, JF. The OPL - Optimization Programming Language. The MIT Press; Cambridge, Massachusetts, London, England: 1999.
13. Fourer R, Gay DM, Kernighan BW. A Modeling Language for Mathematical Programming. Management Science. 1990; 36:5, 519–554.
14. Rosenthal, RE. GAMS - A User's Guide. GAMS Development Corporation; Washington DC: 2015.
15. Assourocko, I., Denno, P. A Metamodel for Optimization Problems. Gaithersburg, MD: 2015. NIST Interagency/Internal report (NISTIR 8096)
16. NIST. NIST Modeling Methodology for Smart Manufacturing Systems. 2015. <https://github.com/usnistgov/modelmeth/tree/master/models/optimization>

Figure 1.
ISA 95 Levels (ISO 2014)

Figure 2.
Information model for the APC-O system [8]

Figure 3.
Information model for the APC-O Variable Type [8]

Figure 4.
Optimization definition Type [8]

Figure 5.
A simplified schematic diagram of the TE chemical process

Figure 6.
Flow of a chemical process optimization based on ISO 15746 standard

Figure 7.
The XML schema for variable

Figure 8.
The XML schema for controlled variable

```

859 <OptimizationModule>
860   <name>Optimization</name>
861   <Type>Optimization</Type>
862   <VendorAttributesOptimizationModule>
863     <!-- [20 lines]
864     <ExpertSystemOpt>
865       <InputConditioning></InputConditioning>
866       <Rules>
867         <Rule>Rg= 0.1*Tr^2</Rule>
868         <InitiationCriteria>
869           On update on change of Tr
870         </InitiationCriteria>
871       </Rules>
872     </ExpertSystemOpt>
873   </VendorAttributesOptimizationModule>
874   <VariableSets>
875     <DependantVariableSet>
876       <Variable> [33 lines]
877     </DependantVariableSet>
878
879     <IndependentVariableSet>
880       <!--<Variable> [44 lines]
881       <Variable> [30 lines]
882     </IndependentVariableSet>
883
884     <StateVariableSet>
885       <Variable> [18 lines]
886       <!--<Variable> [142 lines]
887     </StateVariableSet>
888
889   </VariableSets>
890   <EventSet> [14 lines]
891 </OptimizationModule>

```

Figure 9.
A XML instance for the optimization module.

Figure 10.
The usage of the OPLmetamodel

Figure 11.
OPL code and CPLEX execution screen

Table 1

Information model symbols and definitions

Symbol		Description
object	An object is an item that exists or can exist once constructed, physically or informatically. Associations among objects shall constitute the object structure of the system being modeled, i.e. the static, structural aspect of the system.
aggregation-participation relation link	A fundamental structural relation. Aggregation-Participation is a source item that aggregates one or more other participant items, the destination items, into a meaningful whole.
Exhibition-characterization relation link	A fundamental structural relation. Exhibition-Characterization means that an item exhibits, or is characterized by, another item. The Exhibition-Characterization relation binds a source item, the exhibitor, with one or more destination items, which shall identify features that characterizes the exhibitor.
Generalization-Specialization relation link	Generalization-Specialization relations extend the inheritance concept to both objects and processes. A specialization item has at least the same structural relations and procedural relations as the general item.
Classification-Instantiation relation link	Classification-Instantiation relations connect classes to their instances.