

Integrated management of hierarchical levels: towards a CAPE tool

Canan Dombayci^a, Sergio Medina^a, Moisès Graells^b, Antonio Espuña^{a*}

^a*Chemical Engineering Department, ETSEIB, Universitat Politecnica de Catalunya, 647 Diagonal Avenue, Barcelona 08028, Spain*

^b*Chemical Engineering Department, EUETIB, Universitat Politecnica de Catalunya, Comte d'Urgell 187, Barcelona 08028, Spain*

**antonio.espuna@upc.edu*

Abstract

The integration of decision-making procedures usually assigned to different hierarchical production systems requires the use of complex mathematical models and high computational efforts, in addition to the need of an extensive management of data and knowledge within the production systems. This work addresses this integration problem and proposes a comprehensive solution approach, as well as guidelines for Computer Aided Process Engineering (CAPE) tools managing the corresponding cyberinfrastructure. This study presents a methodology based on a domain ontology which is used as the connector between the introduced data, the different available formulations developed to solve the decision-making problem, and the necessary information to build the finally required problem instance. The methodology has demonstrated its capability to help exploiting different available decision-making problem formulations in complex cases, leading to new applications and/or extensions of these available formulations in a robust and flexible way.

Keywords: Scheduling, Batch Control, Mathematical Modelling, Knowledge and Information Management, Ontologies.

1. Introduction

Much research has addressed integrated management of production systems. However, it is generally accepted that academia and industry do not achieve a common ground for production systems integration since the mapping between the real complex systems and the case studies is hard to accomplish. On the other hand, a considerable number of investigations have recently started to establish a link between the hierarchical levels (Vegetti and Henning, 2015 and Muñoz et al., 2015). An essential element in linking planning and scheduling activities is due to ANSI standards (ISA88, ISA95), which deal with the procedural, physical, and process models, and data exchange in the hierarchical systems. Similarly, supporting systems according to these standards can be found such as BatchML (BatchML, 2015). An extensive study of the general problem of vertical and hierarchical integration is required. Therefore, there is a need to solve this integration problem using advanced tools for supporting the decisions to be made.

The vertical and horizontal integration strategies require extensive formulations. In recent research, three main integration strategies are reported for solving the problem: (i) hierarchical, (ii) iterative, and (iii) full-space methods. All of them require computationally effective formulations, communication between the models and

complementary strategies (Maravelias and Sung, 2009). Recently, Muñoz et al. (2015) used a full-space method in order to solve the integration problem, and used an ontological approach for communication between master and slave formulations in a Lagrangian decomposition approach. Also, Fedorova et al. (2015) used generic model templates constructed with an ontology, providing several layers in order to create the models using existing templates in another domain from this work. In addition to these integration techniques and generic approaches, this work introduces a further step to bring a new solution to the integration of planning and scheduling levels. The solution strategy of decision support for the integrated management system still remains unclear.

This paper addresses integration problems using ontologies and proposes a general solution strategy for making decision in integrated management systems. The starting point is an ontology for planning and scheduling in batch production systems. A domain ontology from the recognized Batch Control Standard ISA88 (Dombayci et al., 2015) is used and enhanced with normative rules. These rules from description logics are added to the ontology in order to maintain the connection between the flows of data from interface to formulations. In this stage, the main idea of the solution approach is to use the same mathematical programming model, and to build new models from this structure supported by the ontology. The details of developed procedure are explained in the methodology section by using separated modules. Additionally, the findings of this study have a number of crucial implications for future practice in decision-making procedures of the integration of hierarchical levels in batch production systems.

2. Methodology

The proposed methodology creates the systematic analysis of the applied methods by categorizing them into modules. An overview of this procedure and its sections are shown in Figure 1 and explained in Sections 2.1, 2.2, 2.3, and 2.4.

The methodology is implemented using markup languages such as OWL (web ontology language) and XML (extensible mark-up language). Additionally, specific tools are used in the implementation of the methodology: (i) a user interface is created using Matlab GUI (graphical user interface), (ii) structures to build XML files are constructed using Matlab, (iii) connection between XML and OWL is achieved with a java-based ontology management tool, and (iv) mathematical formulations are implemented in GAMS.

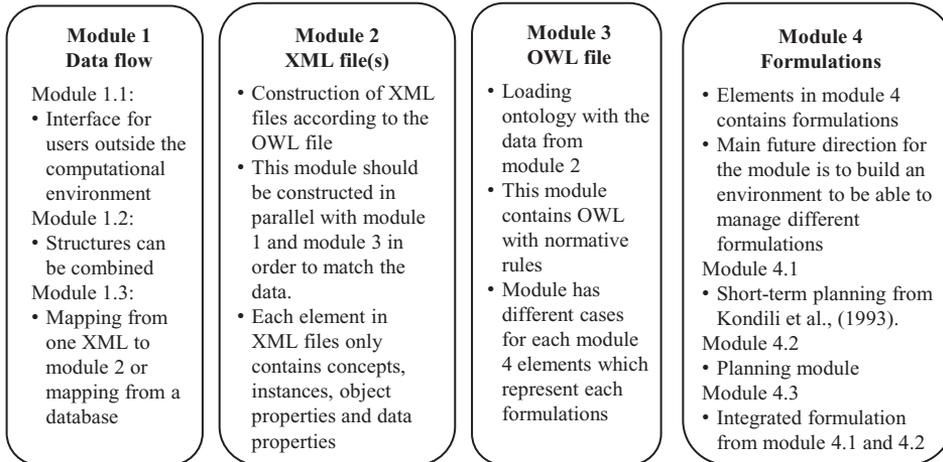


Figure 1. Methodology overview

2.1. Module 1: Data flow

This module is constructed in order to maintain the data flows from/to the different system interfaces. This collects the data based on ISA88 Standard and is developed to sustain the functionality of the work, considering the compatibility with the other modules. Additionally, this module is the place where the decision-making problem is selected. Current implementation is done by Matlab GUI and a drawing tool for STN (state task network). An XML from STN is created from a drawing tool yEd which produces the network as a txt file (Figure 2). Also possible implementations of module 1 are listed as follows:

- Interface for users to build physical, procedural, and process models and/or to load from already created XML files,
- Structure creations without using the interface for module developers, and
- Mapping from previously structured elements from other XML based data such as models, parameters, and data from databases.

2.2. Module 2: XML file(s)

Module 2 contains XML file(s) in order to maintain communication between module 1 and module 3. Each node of XML file contains concept names, instance names, object properties, and data properties to be loaded to the OWL file in module 3. This module is significant for connecting separately structured data and needs to be developed in coordination. XML file(s) are loaded to the OWL file using an instance loader manager based on Java in module 3. The template of XML file is illustrated in Figure 3.

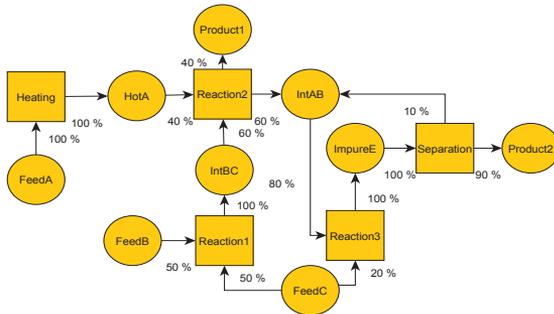


Figure 2. STN of the case study

```
<?xml version="1.0" encoding="utf-8"?>
<cepima>
  <instanceCreator>
    <className>ConceptName</className>
    <instanceName>InsName</instanceName>
    <objectProperty>
      <property>
        <name/>hasObjectProperty
        <value/>OtherInstance
      </property>
    </objectProperty>
    <dataProperty>
      <property>
        <name/>dataProperty
        <value/>someValue
      </property>
    </dataProperty>
  </instanceCreator>
</cepima>
```

Figure 3. XML template

2.3. Module 3: OWL file

Module 3 contains OWL file(s), which have normative rules, modelled according to the each formulation in module 4. This section with OWL files receives the organized instances from module 2 and loads to the static structured OWL file. The static structure means that the concepts and their properties remain the same but instances and their properties are changed in the instance layer by the structured data from the XML file. A substantial part of this module focuses on data management to support the decision-making procedure. For instance, while the connection between the process cell and units are not required to solve a scheduling problem, connections are required to be introduced in the solution of multi-process cell scheduling problem and its planning procedure. While process input refers to material, which is stored or used to a planning formulation, it has different connection with the each unit procedure (or unit) in a scheduling formulation. These connections are maintained by the normative rules, and leading to robust connection between structured data in ontology and input structure of formulations. Table 1 gives the introduced data through the user interface. Data connection between unit procedures and process inputs (STN) is introduced using a

drawing tool. After this, module 3 concludes with the information in Table 2; which shows examples of interaction between the data for each formulation using normatives.

Table 1. Introduced data

Description	Process Input	Process Output	Intermediate Material	Unit Procedure
Entered values	A, B, C	P	AB	UP1, UP2

Table 2. Examples of normative rules and conclusions

Normative rule in developed ontology	Printed Data	Meaning in the formulation
Instances of ProcessInput concept	A, B, C	Raw material set in planning
Instances of UnitProcedure.hasInput	UP1.A, UP1.B, UP2.C, UP2.AB	Input material of unit procedures in order to create mass balance in scheduling formulation
Instances of UnitProcedure.hasOutput	UP1.AB, UP2.P	Output material for mass balance

2.4. Module 4: Formulations

A short-term scheduling formulation (Kondili et al., 1993) is used to maintain the scenarios in results in Section 3. The idea is to use the same formulation and produce different solutions according to the data flow coming from other modules and requirements of the decision-making procedure. The future advancement for this module is to have a general formulation in order to respond the introduced from this methodology and solve the optimization problems.

3. Results

The methodology is implemented as a CAPE tool and different scenarios are used to show the ability of adaptability to other scenarios. The data for solving the scenarios are introduced through the user interface module. Introducing master recipe instances are shown in Figure 4. Scenarios are planned for a multi-process cell area in a site. The decision-making problem is constructed from an area manager point of view and the production planning is supported by finding different scheduling results. In these results, demand is divided for each cell for separate solutions (scenario 1 and scenario 2), demand is added up for all the cells for total solution (scenario 3a), and demand is divided for each cell for a one solution (scenario 3b). Details of scenarios are given as follows:

3.1. Scenario 1

The first scenario is the original problem from Kondili et al. (1993), and is called Process Cell 1 (PC1).

3.2. Scenario 2

The second scenario is similar to the original problem in scenario 1, but max batch sizes are increased and decreased by 10% in Process Cell 2 (PC2) and Process Cell 3 (PC3),

The screenshot shows a web-based interface for defining a Master Recipe. It consists of several input fields and buttons. The fields are:

- Master Recipe ID:** 151010
- Recipe Procedures:** Heating_Reaction1_Reaction2_Reaction3_Sepa (with example text: eg. heating, cooling, mixing)
- Equipment Requirements:** Heater_Reactor1_Reactor1_Still (with example text: eg. reactor1, reactor2, separator)
- Process Input:** FeedA_FeedB_FeedC (with example text: eg. A, B, C)
- Process Output:** Product1_Product2 (with example text: eg. P, E)
- Intermediate Material:** HotA_IntAB_IntBC_ImpureE (with example text: eg. A_cold, AB, G)
- Process Cells:** ProcessCell1

 Below these fields is a 'Save the Master Recipe' button. At the bottom of the interface are three buttons: 'Enter Product Specific Information', 'Enter Unit Specific Information', and 'Enter Process Cell Specific Data'.

Figure 4. Implemented interface view

respectively. Part of the structured data in XML files from scenario 1 (process input, STN) are used directly for these new cells. These additional process cells are solved separately.

3.3. Scenario 3

Scenario 3 has the integrated solution of these 3 process cells and the formulation receives all the input data for the solution. In scenario 3a demand is multiplied by 3 and no specific process cell is addressed for production. In contrast, scenario 3b consists of mapping the amount of demand and process outputs for each process cells according to process cell capacity change. Furthermore, intermediate storage amount is triplet for these scenarios.

4. Discussions

Table 3 summarizes the results for an area manager. Each scenario is designed for different decision-making procedures and is proposed to show the capability of the methodology for complex cases. For instance, if the question is to assign process cells to specific production orders, the manager chooses within scenario 1 and scenario 2 by inspecting the optimal scheduling solutions by checking the make-span time from the table. Scenario 3a contains the optimal scheduling data when the demand is not assigned to specific process cells and scenario 3b considers the assigned demand to each process cells for a more specific situation. Scenario 3a has the highest profit since the problem is constructed monolithic and constraints on demand attendances are removed comparing to scenario 3b. Furthermore, computational times are shown in the table. The required computational effort is lower when the max batch sizes increases (scenarios 1 and 2). Also the required effort is reduced when constraints on product demand are removed (scenario 3a).

Table 3. Result of scenarios

Scenarios	Process Cells	Product	Demand	Make-span	Profit	CPUs
1	PC1	1	500 units	40 hour	13432	0.593
		2	400 units	31 hour	units	
2a	PC2 (10% more)	1	500 units	40 hour	13535	0.453
		2	400 units	21 hour	units	
2b	PC3 (10% less)	1	500 units	40 hour	13393	0.889
		2	400 units	21 hour	unit	
3a	PC1, PC2, PC3	1	3 × 500 units	40 hour	40705	0.125
		2	3 × 400 units	22 hour	units	
3b	PC1	1	500 units	40 hour		0.562
		2	400 units	25 hour		
	PC2	1	550 units	39 hour	40602	
		2	440 units	33 hour	units	
	PC3	1	450 units	32 hour		
		2	360 units	30 hour		

The proposed data management approach recognizes each decision-making procedure in a similar way, however handles data according to the problem to be solved. In this work, an area manager is considered, but this factor can be replaced by a planning formulation and the same data will be shared by two different formulations using module 3. The proposed procedure is to remove the limitations between the hierarchical levels in production systems and to allow the solutions flexibility considering the interactions and decision variables. However, it is necessary to develop and implement different formulations in module 4 in order to solve and improve decision-making problems. In addition, the short-term scheduling formulation is used for different purposes than the reported results for to demonstrate capability of the methodology.

When other related methods and this study are connected, work is in the same research line with Muñoz (2011) in terms of using functionalities of ontological models and the knowledge introducing and collecting methods are managed generally. Additionally, this work is not only on the interfacing of different elements in production systems but also the solutions of mathematical programmes compared with Vegetti and Henning, (2015). Modelling in the ontology is the main issue in this methodology, and similarly connecting this model with programming skills. In general, problems occur in these connections and more difficulties are expected in the solving of large scale problems.

5. Conclusions

This paper proposes a methodology for integrated management of production systems. It also presents a modular approach, and introduces a flexible way of managing production in different process cells while incorporating the planning requirements. The data needed to solve the different optimization problems in different production scenarios are introduced to a general class of problem formulation through a single interface, and the ontology determines the problem instance to be solved. The methodology showed robustness and flexibility for developing more complex cases and may be adapted to use different auxiliary tools (like sophisticated drawing tools to efficiently feed data to the ontology).

Future work in this line involves developing a more general formulation to address other classes of problems in hierarchical systems. Thus, extended formulations should be implemented and the capacity of the methodology should be tested accordingly. Additionally, exploring data base applications to connect module 1 with other modules and investigating further data exchange applications will be investigated.

Acknowledgments

Financial support received from the Spanish Ministry of Economy and Competitiveness and the European Regional Development Fund (research Project SIGERA, DPI2012-37154-C02-01), the ‘Agencia de Gestió d’Ajuts Universitaris i de Recerca-AGAUR’ (2014 FI00305), the Mexican National Council for Science and Technology (CONACyT) and the Research Group CEPEiMA (2014SGR1092), is fully appreciated.

References

- BatchML. Batch Markup Language. Retrived September 2015 from:
<http://www.mesa.org/en/BatchML.asp>.
- C. Dombayci, J. Farreres, H. Rodríguez, E. Muñoz, E. Capón-García, A. Espuña, M. Graells, 2015. On the process of building a process systems engineering ontology using a semi-automatic construction approach. *Comput-Aided Chem. Eng.*, 37, 941–946.
- M. Fedorova, G. Sin, R. Gani, 2015. Computer-aided modelling template: Concept and application. *Comput Chem Eng*, 85, 232-247.
- E. Kondili, C.C. Pantelides, R.W. Sargent, 1993. A general algorithm for short term scheduling of batch operations. *Comput Chem Eng*, 17, 211-227.
- C. T. Maravelias, C. Sung, 2009. Integration of production planning and scheduling: Overview, challenges and opportunities. *Comput and Chem Eng*, 33, 1919-1930.
- E. Muñoz, E. Capon-Garcia, J. M. Lainez-Aguirrec, A. Espuña, L. Puigjaner, 2015. Supply chain planning and scheduling integration using lagrangian decomposition in a knowledge management environment. *Comput Chem Eng*, 72, 52-67.
- E. Muñoz, 2011. Knowledge management technology for integrated decision support systems in process industries. Phd, UPC.
- M. Vegetti, G. Henning, 2015. An ontological approach to integration of planning and scheduling activities in batch process industries. *Comput-Aided Chem Eng*, 37, 995-1000.