

Closing the Information Loop in Recipe-Based Batch Production

Estanislao Musulin, María J. Arbiza, Anna Bonfill, and Luis Puigjaner¹

Chemical Engineering Department, Universitat Politècnica de Catalunya,
Av. Diagonal 67, 08028-Barcelona, Spain.

Rasmus Olsson and Karl-Erik Årzén

Department of Automatic Control, Lund University,
Box 118, SE-221 00 Lund, Sweden.

Abstract

In addition to the basic regulatory functions, a batch control system must support production planning and scheduling, recipe management, resource allocation, batch report generation, unit supervision and exception handling. A closed-loop framework is presented in this work that integrates decision support tools required at the different levels of a decision-making hierarchical batch control system. Specifically, the proposed framework consists of a reactive batch scheduler (MOPP) and a fault diagnosis system (ExSit-M) developed by the Universitat Politècnica de Catalunya, and a S88-recipe-based coordinator (JGrafchart) developed by the Lund University. These tools need to exchange information to obtain optimal utilization of the production plant. The complete integrated system is built using a general recipe description and other guidelines from ISA S88 standard (ANSI/ISA 1995).

Keywords: Batch, Integration, Reactive Scheduling, Fault Diagnosis, Recipe

1. Introduction

In a production plant environment the presence of unpredictable events not only related to external market factors but also to the operational level, e.g., equipment breakdowns and variable operation times, is usually unavoidable. Despite the uncertainty in the production scenario, the scheduler has to make some decisions both to start production and to face abnormal events. The need to increase the reliability of any decision-making process, thus reducing the gap between theory and practice, makes necessary to take this uncertainty into account.

Research in scheduling under uncertainty has mostly been focused either on rescheduling algorithms, which are implemented once the uncertainty is disclosed, or stochastic approaches that incorporate the uncertain information at the decision level prior to scheduling. On one hand, the execution of deterministic optimal schedules based on nominal parameter values and the implementation of rescheduling strategies to tackle the problem once the uncertainty is revealed can result cumbersome or unrealistic without previous consideration of the uncertainty. If the uncertainty can be characterised at the time of scheduling, it should be advantageous to take possible future events into consideration before they happen in order to minimise the negative

¹ Author to whom correspondence should be addressed: Luis.Puigjaner@upc.es

outcomes. On the other hand, the future cannot be perfectly forecasted so, despite considering the uncertainty a priori, deviations from the predicted schedule can always occur once the uncertainty is realised. Therefore, it is required to adapt the schedule to the new scenario if a good performance of the system is pursued.

The integration of a Fault Diagnosis System (FDS) aims to timely provide the process state information to the different levels in the decision-making hierarchical structure, thus reducing the risk of accidents and improving the efficiency of the reactive scheduling in the most effective way.

To handle unit supervision, exception handling, and recipe execution a coordinator is implemented in JGrafchart. The unit supervision is based on modelling the state of each equipment object and procedural element using finite state machines.

A closed-loop framework for on-line scheduling of batch chemical plants integrating, robustness considerations, fault diagnosis, recipe coordination, and exception handling is proposed in this work. This on-line integration leads to a fast execution of the recovery procedures and the rescheduling.

2. Scheduling and reactive scheduling

The developed scheduler uses the Event Operation Network (Cantón 2003) to model the system and has a library of different dispatching rules to determine a feasible schedule. The dispatching rules available can be classified into three sets: priority rules that determine a list of recipes to be sequenced and assigned to specific units, assignment rules that determine which equipment should be used for each stage of each batch, and sequencing rules that determine the sequence of batches and the sequence of operations for each unit. It also has a library containing a variety of heuristic and rigorous optimization algorithms to determine an initial optimum schedule. Furthermore, the objective function used by the optimization algorithms can be customized to optimize the use of resources, cost of changeovers, profit, makespan, etc. Once generated, the optimum schedule is sent to the coordinator to be executed in the process.

Unexpected events or disruptions can change the system status and affect its performance. Therefore, during the on-line execution the scheduler receives from the process coordinator information about the actual executed schedule. Deviations from the original schedule and information about equipment breakdowns coming from the FDS will trigger a rescheduling (Arbiza et al. 2003 and Bonfill et al. 2004). The new generated schedule will be optimum according to the new plant situation. If some modification is made, the new schedule is sent to the process coordinator. The rescheduling algorithm (Arbiza et al. 2003b) is presented in Table 1.

Table 1: Rescheduling algorithm

-
- 1 - Create a master schedule.*
 - 2 - Send schedule to the process coordinator.*
 - 3 - Receives the actual executed schedule from the process coordinator.*
 - 4 - Generate new optimal schedule.*
 - 5 - If the new schedule differs from the implemented one go to 2, else go to 3.*
-

The rescheduling system is completely configurable and customizable considering the manager objectives. It allows selecting different dispatching rules, optimizers and objective functions according to the process knowledge. The alternative rescheduling

techniques (recalculate a new robust schedule, recalculate schedule without robustness considerations, actualize operation times, reassignment, etc.) are evaluated and the system selects the best suited ones according to the objective function adopted. Optimization algorithms may be included depending on the interest of the decision maker and the required reaction time.

3. Fault Diagnosis

The FDS is designed based on artificial neural networks (ANN) and fuzzy logic, with a modular structure based on process decomposition following the ISA S88 standard. It was developed using G2[®], and operates in collaboration with the coordinator and the scheduler sending complete information with regard to the process state (equipment breakdowns, time of unavailability, etc.). Furthermore, it incorporates a complete decision-support system for the process operator based on the information obtained from a HAZOP analysis and a user friendly graphical interface.

Normal operation conditions modelling is a central issue in batch process monitoring. To improve and simplify the modelling a step-wise model of the process is built. Each unit is represented by a set of ANN models that model the behaviour of each unit during a specific operation. In processes with complex dynamics this step-wise modelling can be extended to model the equipment at the phase level. Then, during the on-line operation, a model-manager activates and deactivates the models depending on the active process operations that are being executed into the process; information that comes from the coordinator.

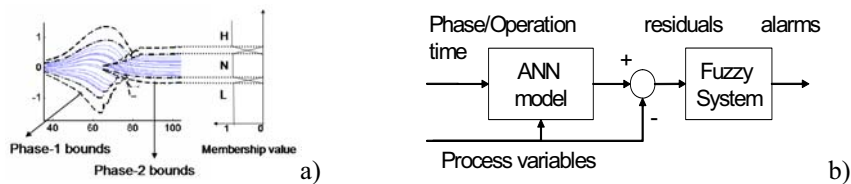


Figure 4: Fault Diagnosis system. a): Residual bounds and membership functions definition in the intersection of two control phases. b): Partial equipment model with its diagnosis stage.

The model predictions are compared with historical data to obtain limits for the normal operation conditions. Residuals corresponding to one variable from 20 operation runs are presented in Figure 4a. The area between the inner limits is considered as the normal behaviour region. Outer limits are calculated by multiplying the inner bounds by a factor. The factor depends on the trade-off between incipient fault diagnosis and robustness (no false alarm generation). Note that the limits depend on the process variability along the operation/phase time, and using the phase-time instead of the operation-time the limits can be set tighter especially around the change of phases. Finally, the methodology presented in (Ruiz et al. 2001) has been extended to obtain rules from a HAZOP analysis. Rules are introduced into a fuzzy system to relate the deviated residuals with faults. The membership functions change during the operation in such a way that residual values in the inner limits are considered normal, values located between the two limits lie in two fuzzy sets (High and Normal or Low and Normal), and finally, values located outside the external limits are considered to lie either in the Low or High set (Figure 4a). For each control operation the system shown in Figure 4b is applied.

4. Coordinator

The coordinator is implemented in JGrafchart, a Java implementation of Grafchart (Årzén 1994). The coordination involves management of scheduled batches, recipe execution, unit supervision, alarm propagation, and exception handling. The plant is divided into units according to S88. Each unit consists of equipment and control modules such as agitators, valves, and pumps. The units also contain the equipment control. The recipe/equipment control separation is on the operation level in S88, i.e., the recipe makes a procedure call from a procedure step representing a recipe operation to a procedure representing the corresponding equipment operation.

Within the coordinator each batch is represented by a control recipe expressed using Sequential Function Chart (SFC) formalisms. Since the control recipe is carrying all the information about the development of the batch a report can be sent back to the scheduler every time a new phase is started. If an exception occurs and a batch has to be aborted this information is also sent back to the scheduler.

The unit supervision is based on finite state machine models of the state of each equipment object and procedural element (Olsson 2002). The equipment state machine serves two purposes. The first is to be able to check that all the equipment objects are in a consistent state when an operation is invoked. The second purpose is to provide a structure for organizing the safety and supervision logic at the equipment control level. If a fault occurs, the safety logic causes a state transition from a normal state to a fault state. The state of an equipment/control module will propagate up to the unit level.

Most of the functionality is associated with equipment operations. Each equipment operation object contains a procedure (i.e. the sequential control) and a state machine monitoring the state of the procedure. The procedure of an equipment operation holds not only the equipment sequential control, but also contains several checks, which need to be performed when a procedure is called from a recipe. It checks if the procedure itself is in the Idle state and, if so, changes the state to Running. The check if the unit is in a consistent state at the start of the operation is also checked using state machines.

The separation between the normal recipe execution and the exception handling can be made in different ways. In Procel most of the exception handling is operation specific. When a procedure containing the operation is called the associated exception handling is enabled. The exception handling logic of an operation involves both the recipe level and the equipment level. Exception handling logic that must be active also for an idle equipment unit is contained in the unit exception handling object.

Exception handling is also needed at the recipe level. For example, an exception that has occurred must be fed back to the control recipe, recorded in the batch report and sent to the scheduler, and appropriate actions must be taken to deal with the exception.

An important consideration is how to separate the recipe information from the exception handling logic and operations. The actions that are taken in the recipe depend on the type of exception. In a few special cases it might be possible to “undo” an operation and rollback the execution of the recipe to a safe execution point, and from there continue the execution using, e.g., a new unit. However, due to the nature of chemical batch processes a rollback is in most cases not a viable alternative. Also in the more common case where the batch cannot be produced as intended there are several alternatives. In certain situations it might be possible to still make use of the batch to produce a product of a different quality. In other situations it is possible to recirculate the batch ingredients for later reuse.

5. Integration Methodology and Technology

The proposed integrated framework along with the flow of information through the different modules is depicted in Figure 1. There exists a central agent (DTM) that manages the information flows.

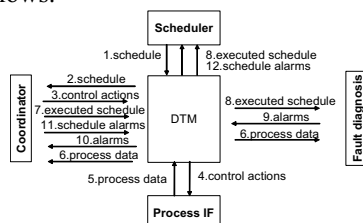


Figure 1: Integration diagram

The scheduler generates an initial schedule that is then translated into control actions and executed into the process by the Process Coordinator. When an abnormal event is detected by the fault diagnosis system (FDS), it sends an alarm to the scheduler through the Process Coordinator, which executes some pre-specified recovery procedure, depending on the alarm. The scheduler receives the alarm and generates a new optimum schedule. All the information is stored in an ISA S88 compliant database.

The developed software toolboxes have been integrated in a common infrastructure, named the CHEM Communications Manager (CCOM) (CHEM 2003) that allows communication through the exchange of XML messages. It is based on public domain Message Oriented Middleware (MOM) software that provides Publish/Subscribe and Point to Point message communication. CCOM acts as a server that clients can connect to. Moreover, a client API has been developed on top of the MOM interface to provide additional functionality and hide the aspects of transport protocols to the clients.

6. Case study

The proposed integration architecture has been successfully tested on PROCEL, a pilot plant located at UPC (Fig. 5a). It consists of three tank reactors, three heat exchangers, sensors, and the necessary pumps and valves to connect the equipment.

Tests of the performance of the FDS, and the reaction of both the coordinator and the scheduler in case of abnormal events have been performed. A test starts with the generation of an initial schedule and its execution into the plant. During the execution of the schedule a fault is introduced. The FDS isolates the fault and informs the coordinator about the equipment unavailability. The coordinator starts an exception-handling procedure to abort the batch and sends a schedule alarm to the scheduler. A new schedule considering the actual plant situation is generated and sent to the coordinator for its execution. Once the fault is corrected, the loop is repeated to find a new optimum schedule considering the repaired equipment. In Figure 5b, the main GUI interface of the scheduling package is presented. It summarizes the execution of the test. The upper left of the screen shows a Gantt-chart of the initial schedule. The down left part shows the actual executed schedule. There is a dotted batch that represents a faulty batch. Finally, at the upper right is presented the new schedule.

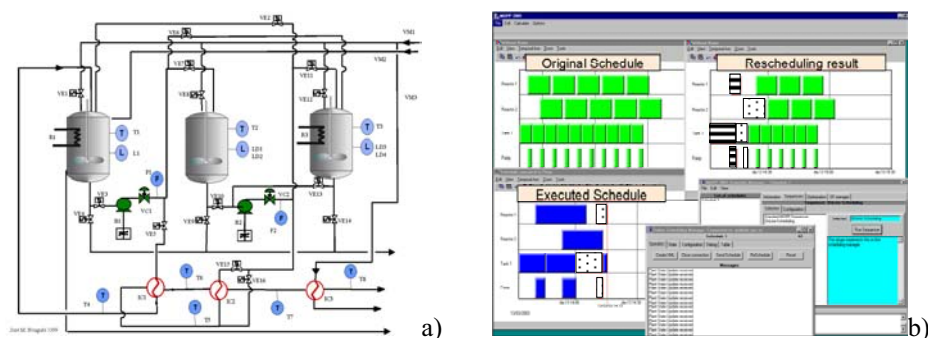


Figure 5. Case Study: a) Flexible chemical plant (PROCEL) flowsheet. b) Interface of the scheduler showing the original scheduler, the actual schedule and the rescheduling proposed.

7. Conclusions

The applicability and effectiveness of the proposed on-line integrated framework is illustrated with its implementation into a batch chemical plant. The integrated system has shown the ability of detecting and reacting to abnormal process events under uncertainty. A structured FDS approach has been presented that leads to simpler, robust and faster to train models, which allow tighter detection limits leading to an incipient and robust detection of faults. Knowledge from a HAZOP analysis is introduced as rules to isolate the faults and to support operator decisions. The simplicity and adaptability of the FDS for its application in complex plants is presented. An open and flexible system for rescheduling has also been presented which takes advantage of user's process knowledge. The efficiency of the rescheduling system to adapt the schedule to the current situation in the plant has been successfully tested.

Acknowledgement

Financial support from the E.C. (Project G1RD-CT-2001-00466) is gratefully appreciated.

References

- ANSI/ISA, 88.01 Batch Control, Part 1: Models and Terminology, 1995.
- Arbiza M.J., Cantón J., Espuña A., Puigjaner, L. *Flexible rescheduling tool for short-term plan updating*, AIChE 03', San Francisco, USA, 16-21 November 2003.
- Arbiza, M.J., Cantón, J. Espuña, A. and Puigjaner, L. *Objective based schedule selector: a rescheduling tool for short-term plan updating*, [CD-ROM]. ESCAPE 14, Lisboa, 2003b
- Årzén, K.E. *Grafset for intelligent supervisory control applications*. Automatica, Volume 30, Issue 10, October 1994, Pages 1513-1525.
- Bonfill A., Arbiza M.J., Musulin E., Espuña A., Puigjaner, L. *Integrating robustness and fault diagnosis in on-line scheduling of batch chemical plants*. In: Proceedings of IMS International Forum 2004, Milano, Italy, pp. 515 – 522.
- Cantón J., 2003, *Integrated support system for planning and scheduling of batch chemical plants*, PhD Thesis, Universitat Politècnica de Catalunya, España.
- CHEM, *Advanced Decision Support System for Chemical/Petrochemical Manufacturing Processes*. [On-line][Accessed 2003] Available on: < <http://www.chem-dss.org/>>.
- Olsson, R. *Exception handling in recipe-based batch control*, Licentiate thesis, Department of Automatic Control, Lund Institute of Technology, 2002, Sweden.
- Ruiz, D., Nougues, J.M., Puigjaner, L. *Fault diagnosis support system for complex chemical plants*. Computers & Chemical Engineering, 25, pp. 151-160 (2001).