REACTIVE SCHEDULING IN REAL-TIME OPERATION OF BATCH CHEMICAL PLANTS

Jordi Cantón, Jose María Nougués, María Jose Arbiza, Ricard Gonzalez,
Antonio Espuña and Luis Puigjaner
Chemical Engineering Department, Universitat Politècnica de Catalunya
Av. Diagonal 647, 08028- Barcelona (Spain)

Abstract

The aim of this work is to show how the integration of the planning level with the co-ordination level of the CIM architecture have been carried out, in order to close the information loop between a co-ordination module of the batch plant and the scheduling system. The information loop starts at the scheduling tool, which generates an initial schedule, and then the schedule is used by the co-ordination system to automatically perform the necessary actions over the plant. The information about the execution of the real schedule is reported to the scheduling tool and monitored to find possible deviations from the predicted schedule. If an abnormal situation is detected a rescheduling procedure is performed in order to generate a new scheduling according to the updated situation of the real plant.

Keywords

Scheduling, ISA SP88, integration, coordination system, EON, GCO

Introduction

Industrial manufacturing processes can generally be classified as continuous, discrete parts manufacturing, or batch. How a process is classified depends on whether the output from the process appears in a continuous flow (continuous), in finite quantities of parts (discrete parts manufacturing), or in finite quantities of material (batch).

Usually, in the food process industries there are plant sectors that operate in continuous modes and others that work in batch and semicontinuous modes. From the control point of view, this fact has great impact in the design of a plant-wide control system and also in the information flows through the plant. The integration and closure of information loops are key objectives for the flexibility and optimum operation of the plant.

In the past and even today control systems are usually vendor specific solutions. They are offered as complete packages where the user has hardly a possibility to integrate his own software solutions. Users therefore heavily depend on the willingness of control vendors to implement required software extensions.

However, the situation is changing. Process and discrete manufacturing industries need more flexibility, the upcoming of new technologies like computer networks and protocols, and the wish to unite activities. In response to this trend control software companies have put increased emphasis on controlling the process while controlling the changes (Johnson ,1997)

Nowadays, the tendency to solve the problem of process integration, moreover to be aware of design aspects, is automatize in a wide sense the plant to integrate every element of the plant operation, including process control, scheduling and closing the information circuits (closure information loops) inside the plant.

Integration

The concept of CIM (Computer Integrated Manufacturing) (Harrington, 1973) is the result of the efforts to solve the problem of systems integration in the industry. It was developed for mechanical industry (pieces manufacturing, discrete processing), and applied to the process industries is CIP (Computer Integrated Processing) (Gang and Nyberg, 1999). The architecture of a CIM/CIP environment is made up by the following components: Business Scheduling Systems, Production Management Systems and Process Control.

Integration of these components requires a computing support system that includes sensor hardware and actuators communication, network, computer hardware and software. It showed be obvious to conclude that to achieve integration is necessary the existence of not only well defined standards in the information management systems of the plant, but also open information systems in software and hardware.

In the proposed approach, the planning and scheduling system and the control system share a common database that contains the recipe descriptions and plant description data. The most detailed level of recipe information is used mainly for the control system whilst the planning and scheduling system use the less detailed information to perform its task. The advantage of this common database is that a common interface can be used to perform input data, so data consistency between the two modules is ensured. Additionally, the communication between the Planning system and the Control system is carried out using a data server, which performs the queues of the different messages and delivers them on request. The use of a data server allows easy upgrades and changes of the different modules. It also allows performing an easy monitoring of the communication between the different modules of the whole system. Another advantage of this system is its open architecture that allows communicating with third parties software and hardware.

Communication

The communication system is the only mean of the system platform for information interchange between system modules. It supports both, the exchange of information between modules located on the same processor-board as well as also between modules located on different boards connected through a bus-system (network). A standardised protocol has been defined to ensure uniform data formats and a fixed set of messages.

To connect the modules over the network a core bus (see Figure 1) was defined based on the OSI base reference model. It implements any middleware technology from simple TCP/IP protocol or other higher level internet protocol like HTTP, or distributed object technologies like CORBA, RMI or DCOM. Over this middleware technology two additional layers have to be built, one for object mapping specific for the selected middleware technology and other for message parsing. The interprocess communication is made through DDE/OLE technology, and RPC, the same parsing mechanism is used to processing the cross application message. In the figure the system modules are shown over the deployment network, also, in these figure the information transmitted is indicated.

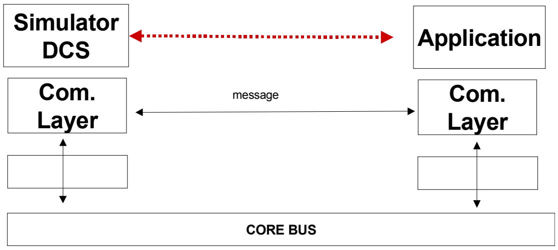

Figure 1. Network core bus.

Scheduling system

The planning and scheduling system uses event operation networks (EON) modelling system (Graells, Cantón, Peschaud and Puigjaner, 1998). The EON model has proved to be an efficient way to describe time constraints between operations in a complex production structure, as it is the batch process. The EON model is built using a general recipe description and other guidelines from ISA S88 specification.

In the first step, according to the present situation of the plant and the client orders, a first batch sequence is generated using the information provided by the recipe and the stage levels. Then, an EON graph is generated using the information located at the operation level description of the recipes, and the information generated in the previous step about the unit assignments and task sequence. Finally different methods can be used to adjust the proposed solution under the constraints imposed by the different resources required.

Just as soon as the schedule is built, the predicted list of operations is available to be used by the coordination system. This information can be easily translated into control recipes to be executed in the coordination control.

Once the schedule is running on the real (or simulated) plant, the coordination system communicates to the planning system the deviations detected from the proposed plan. With this information, the scheduling system performs the rescheduling and the new schedule is sent to the data server. This feedback closes the loop between the planning and scheduling system and the control system.

One important aspect in on-line rescheduling is the time window involved in the whole process as it is shown in Figure 2.

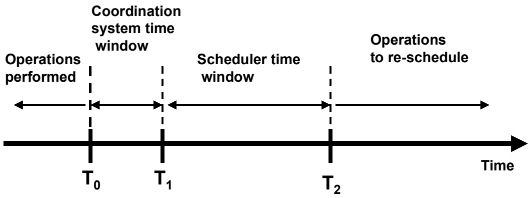

Figure 2. Time windows involved in the rescheduling

At any time (T_0) there is a set of operations already done, which have been reported by the coordination system, as it has been explained before. When a rescheduling process is started at time T_0 there are two aspects to be included in the rescheduling strategy: the coordination system time window and the scheduling system time window.

The coordination system uses a time window to perform its functionality. This means that all the operations planned in the time interval from T_0 to T_1 are responsibility of the coordination system, therefore these operations can not be rescheduled. Any necessary changes in the assignment or timing of the operations scheduled in the coordination system time window should be ordered by the coordination system.

Furthermore, the rescheduling process requires itself a certain period of time to perform the necessary calculations. This required time is the scheduler time window. The scheduler time window, although that it is considerably shorter than the average processing times, should also be taken into account as the time window between T_1 and T_2.

The operations that have already been scheduled within the time interval T_0 to T_2 have the frozen state which mean that can not be rescheduled.

The main reason of setting the frozen state is that when the rescheduling procedure (started at T_0) finishes, the coordination system will be working with a time window that ends at T_2. It has no sense to reschedule operations that may have been finished when the rescheduling procedure has been concluded. In the same way, any new operation should be scheduled after T_2.

Co-ordination system.

The Figure 3 shows the information exchange between levels and also the physical reference model of the S88 has been included to show where and what type of information does the modules manage and how it is transformed. It can be seen the role of the coordination system and where the major co-ordination activities are involved. The key role of the co-ordination module is the recipe explosion, that mean how the control recipe generated by the scheduling module at the operation level is exploited in phases, thus a control recipe at phases information is generated (Nougués et al., 2001).

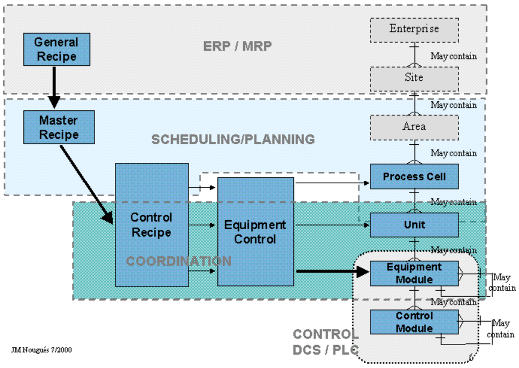

Figure 3. Role of the coordination module.

In the figure 4 it is shown the deployment of the modules over a network and how the recipe is exploited based on the information received from the scheduling and planning level, the current plant status and the recipe information at the phase's level. The information is maintained in two lists one with information at the operation level and other with the phase information (see Figure 5). During the plan execution the co-ordination module reads the information from the scheduling and planning module and exploits the recipe for a predefined time window, taking into account that all the phases of the operation are completely read in each time window. When the end phase condition is reached the system change the status of the phase as completed and if this phase also end the operation mark also the operation is completed in the operation list.

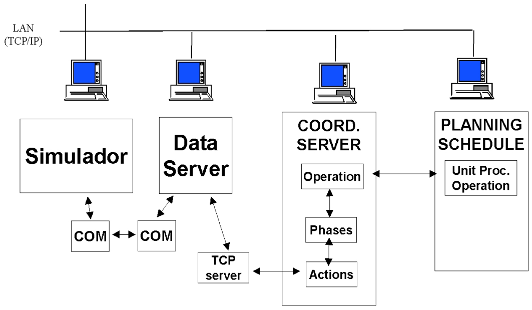

Figure 4. Software modules in the deployment scheme

After a predefined cycle time, the co-ordination writes the status of the operation in the database and when an specified condition occurs sends an event through the network to the scheduling and planning system. For example if an operation can not be completed an event occurs and is sent to the scheduling and planning module. The event messages are coded as raw XML formatted information. The co-ordination module also acts as data server and exposes an interface over the network that allows different type of clients to consult the status of the batch execution. The module acts as server and implements different type of distributed object technologies, CORBA, JAVA/RMI and COM, this allows clients running in different platforms (Windows, Solaris or Linux). The co-ordination module is implemented in JAVA thus also allowing the system to run in different platforms. Figure 5 shows the detailed information flow between the different modules.

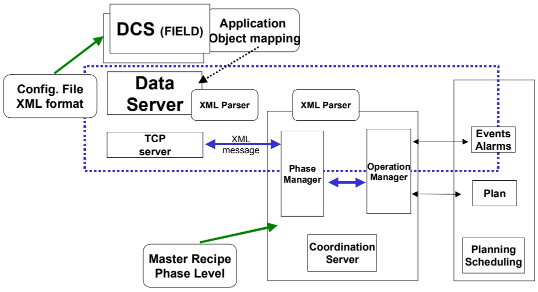

Figure 5. Modules information exchange.

Test case.

For a systematic study of these systems a case study was developed and implemented in a real pilot plant scenario.

Three tank reactors, three heat exchangers and the necessary pumps and valves to allow the changes of configurations constitute the scenario. This physical system allows to validate the proposed methodologies with a simple case study so that they can be transported to more complex production structures (i.e. sugar refineries in food industries). The test problem includes the three layers: planning and scheduling, co-ordination operation, supervisory control. The plant flowsheet is shown in the figure 6. As can be seen the plant can operate in several operations mode continuous, semicontinuous and batch. It is also possible to take out one heat exchanger from the flowsheet opening and closing two valves. The plant has also been simulated with an in house simulator to test the software prototype of the integration architecture.

One aspect of this implementation was to check if a JAVA implemented co-ordination system was capable to execute a batch recipe and control a batch plant. In this aspect the co´-ordination module performs its function without any problem. A key factor to this success was the small amount of computer power needed to perform the sequential batch control in the real plant due the low sampling time used (5 seconds) that requires the transitions to be evaluated only twelve times every minute.

Conclusions

This work has shown an integrated system that performs reactive scheduling in batch chemical plants. The pilot plant scenario demonstrates that effective on-line scheduling is possible taking into account three factors: A common database to assure the data consistency, an adequate communication system to exchange data between the scheduling system and the co-ordination control and, finally, the use of time windows to take into account the necessary calculation time needed in the scheduling system.

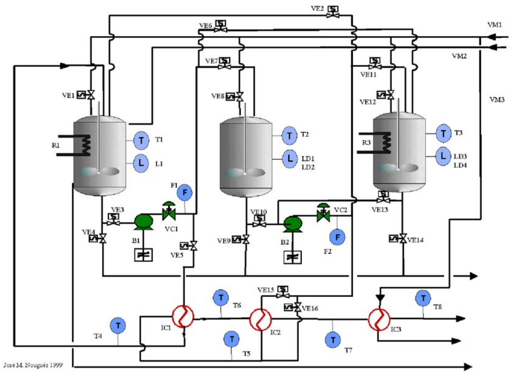

Figure 6. Pilot Plant Flowsheet

Acknowledgements*:*

Support received from the CHEM Project is fully appreciated. CHEM Project is funded by the European Community under the Competitive and Sustainable Growth programme of the Fifth RTD Framework Programme (1998-2002) under contract G1RD-CT-2001-00466

References

Graells, M., Cantón, J., Peschaud, B. & Puigjaner L. (1998). General approach and tool for the scheduling of complex production systems. *Computers & Chemical Engineering*, **22S**, S395-402.

Harrington Joseph*,* Computer Integrated Manufacturing, *Industrial Press*, 200 Madison Av., New York, 1973.

Johnsson C. and K. Arzen, "Grafchart for recipe-based batch control", *Computers and Chem. Engng* **22**, 12 pp 1811-1828, 1998.

Johnson Dick, Mixing Up a Batch!, *Control Engineering*, October, p 68, 1997.

Nougués J.M, González R., Puigjaner L., Coordination Module for Integration Scheduling System with Batch Process Control, paper 85c*, AIChE Annual Meeting,* November 4-9, Reno, NV, 2001.

S88.01 Batch Control, Instrument Society of America (ISA), 1995.

Xiong Gang, Timo R. Nyberg, Comparison between CIMS and CIPS, *7th IEEE International Conference on Emerging Technologies and Factory Automation*, October 18-21 , Barcelona, Spain, 1999.