A Case study: Multi product batch plant for the demonstration of control and scheduling problems

Nanette Bauer*, Stefan Kowalewski, Guido Sand, Thomas Löhl
Process Control Laboratory
Department of Chemical Engineering,
University of Dortmund
Phone: ++49-231-755 5171, Fax: ++49-231-755 5129
E-mail: n.bauer@ct.uni-dortmund.de

Abstract

In this paper a new laboratory plant which was designed and built at the Process Control Laboratory of the University of Dortmund is presented as a case example. The purpose of the plant is to serve as a test-bed and a demonstration medium for control and scheduling methods for multi product batch plants. A scheduling problem is proposed and the control architecture integrating both control and scheduling aspects is described.

Keywords: case study, batch processes, scheduling, control architecture

1 Introduction

The plant we present is Case Study 7 in the European Project *Verification of Hybrid systems*[1] [1]. It was designed and built at the Process Control Laboratory of the University of Dortmund and is a multi product batch plant which produces two products from three raw materials. Compared to batch plants of industrial scale it is still of moderate size although it is already complicated enough to pose complex scheduling and control tasks. To test the schedules generated for this plant, e.g., by a scheduling tool, a control routine is needed to automatically run given schedules on the plant. The control architecture for the plant therefore needs to be suited for scheduling tasks. We here describe a concept for running a schedule on the plant, which considers that processes may be too late but nevertheless are processed in the order which is given by the schedule. The paper is organized as follows: First, the plant is described in Section 2. The reaction taking place is explained and the plant layout is described. In Section 3, a scheduling problem for this plant and a solution to this problem is given. The control architecture is explained in Section 4. The hierarchical structure of the control program is shown, the control tasks are explained and the concept for running a given schedule on the plant is presented. The section 5 gives conclusions and ideas for further research concerning this case study.

2 The Process

2.1 Reaction

The process under consideration is a batch process in which two liquid products, one of blue colour, one green, are produced from three liquid substances coloured yellow, red and colourless. For the purpose of demonstration, the change of the colours when the raw material substances are mixed represents the "production step".

The chemistry behind the change of colours is a neutralization reaction of diluted hydrocloric acid (HCl) with diluted sodium hydroxide (NaOH). The diluted hydrocloric acid is mixed with two different pH indicators to let the acid look yellow, if it is mixed with the first indicator and red when getting mixed with the second one. During the neutralization reaction the pH indicators change their colours when a certain pH value is reached (which is about pH 7), the first indicator changes from yellow to blue, the second one from red to green. Table 1 shows the different colours for the different substances and pH values. In the following, we will refer to the substances only by their colours (the colourless NaOH will be represented by "*White*").

	raw material	**product**
Indicator 1	yellow	blue
Indicator 2	red	green

Table 1: Colours of raw material and the corresponding products

2.2 Plant description

Basically, the plant consists of three different layers, which become apparent in the P&I diagram, Figure 1:

- **Buffer tanks for the raw material substances**: The upper layer consists of the buffering tanks B11, B12 and B13 for the raw materials "*Yellow*", "*Red*" and "*White*", respectively. Notice that each tank is used exclusively for one raw material and may contain at most two batches of substance.

*author to whom correspondence should be addressed

[1] ESPRIT-LTR Project 26270 VHS

- **Reactors**: On the next layer, three reactors R21, R22 and R23 are situated. Each reactor is equipped with a stirrer. As can be seen in the Piping and Instrumentation diagram, Fig. 1, each reactor may be filled from each raw material buffer tank. This means, that it is possible to produce either "*Blue*" or "*Green*" in each reactor. The production is done by first filling one batch of acid ("*Yellow*" or "*Red*") into the reactor and then neutralizing it with one batch of "*White*". Each reactor may contain one batch of product (resulting from two batches of raw material).

- **Buffer tanks for the product**: From the reaction layer, the product batches are emptied into the buffering tanks B31 or B32, which again are exclusively used for "*Blue*" or "*Green*". Both of the tanks may contain at most three batches of product. There are two different options for draining the tanks: First, forced by gravity, then the outflow rate has a nonlinear characteristic, or second, using the pumps thus realizing a constant rate of output. For the first option, only valve V311 for tank B31 (or V312 and H312 for tank B32) have to be opened, for the second, additionally the pumps P4 or P5 have to be switched on.

Hidden from the observer are the tanks B41 - B44 and B51. The tanks B41 - B44 are used for the preparation of the raw material solutions from concentrated hydrocloric acid or sodium hydroxide, tap water and indicators. During plant operation, batches of the raw materials are pumped into the respective tanks via pump P1, P2 or P3, respectively.All tanks are equipped with an overflow pipe going to tank B51. (Tank B51 may be emptied via P5 by closing the manual stop cock H312 and opening manually stop cock H511).

2.3 Processing times

The processing times presented in Table 2 were determined under the following conditions:

- One batch of raw material is 850 ml.

- The initialization of the plant is not considered, times here may differ e.g. because the pumps are pumping air at the beginning etc.

Notice that the neutralization takes place (and does not need additional time) while the sodium hydroxide is drained into the reactor which at this time already contains a batch of acid.

Processes	**Times**
Pumping 1 batch "*Yellow*" into B11	12 sec.
Pumping 1 batch "*Red*" into B12	12 sec.
Pumping 1 batch "*White*" into B13	12 sec.
Draining 1 batch "*Yellow*" into R21	15 sec.
Draining 1 batch "*Red*" into R21	11 sec.
Draining 1 batch "*White*" into R21	10 sec.
Draining 1 batch "*Yellow*" into R22	12 sec.
Draining 1 batch "*Red*" into R22	13 sec.
Draining 1 batch "*White*" into R22	9 sec.
Draining 1 batch "*Yellow*" into R23	12 sec.
Draining 1 batch "*Red*" into R23	14 sec.
Draining 1 batch "*White*" into R23	13 sec.
Draining 1 batch "*Blue*" from R21 into B31	12 sec.
Draining 1 batch "*Green*" from R21 into B32	13 sec.
Draining 1 batch "*Blue*" from R22 into B31	12 sec.
Draining 1 batch "*Green*" from R22 into B32	12 sec.
Draining 1 batch "*Blue*" from R23 into B31	12 sec.
Draining 1 batch "*Green*" from R23 into B32	12 sec.
Pumping 3 batches "*Blue*" out of B31	30 sec.
Pumping 3 batches "*Green*" out of B32	30 sec.

Table 2: Processing times

3 THE SCHEDULING PROBLEM

3.1 Problem formulation

The plant was designed with the intention to provide a variety of scheduling problems. Here, we propose two specific formulations:

Given a particular schedule of delivery times for the batches of "Yellow" and "Red" and "White" schedule the production of "Blue" and "Green" such that

1. *a continuous outflow of the buffer tanks B31 and B32 is realized.*
   This of cause implies that the buffer tanks must not run empty. An additional constraint is that they (as all the other tanks too) must not overflow.

2. *customer demands are fulfilled.*
   These demands consist of a certain amount of product in the respective product tank at given time points, called due dates. For this problem we present an example solution in Sec. 3.2.

The degrees of freedom for the scheduling for both formulations are:

- The dates at which one batch of "*Yellow*" or "*Red*" is emptied from the respective buffer tank into a reactor.

- The waiting time in the reactor. The reactors may be used as buffers for the acid. After the reaction, there is no waiting time but the product is drained into the respective product buffer tank immediately.

For running the schedule on the plant with the help of a control program (4.3), we assume, that the schedule has the structure illustrated in Table 3. Each entry (batch), which in the following we will denote as a *process* consists of a starting time, information about the recipe and the waiting time in the reactor (only for production

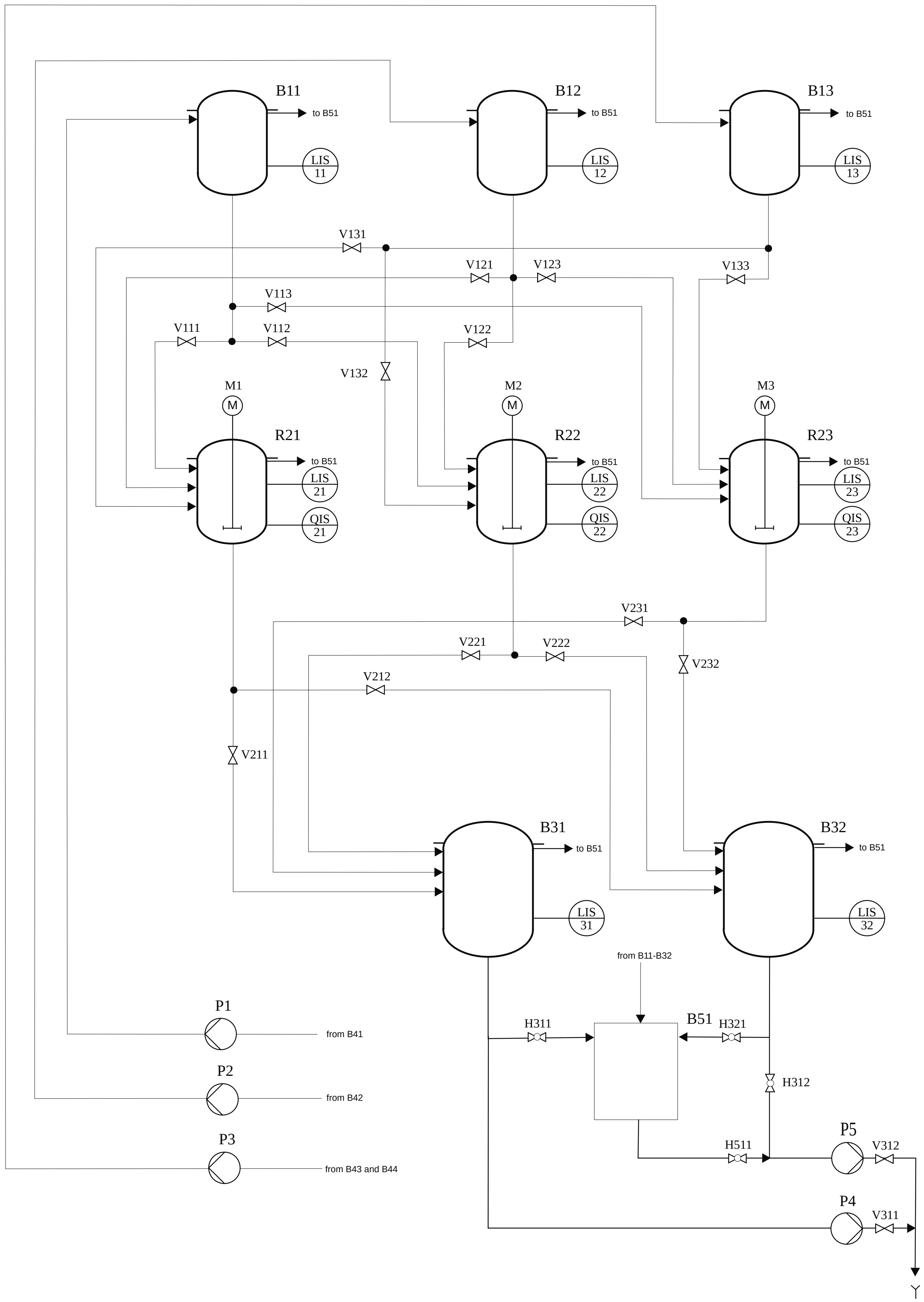

Figure 1: Piping and instrumentation diagram of the demonstration plant

recipes). The information about type, substance and the assigned resource define altogether 9 recipes, 3 deliver recipes (each raw material in a fixed tank) and 6 production recipes (each product in one of the reactors).

No	Start t	Type	Subs.	Res.	Wait t
1	00:00	Deliver	White	B13	-
2	00:30	Deliver	Red	B12	-
3	01:40	Produce	Green	R21	30 sec.
4	01:50	Deliver	White	B13	-
5	01:55	Deliver	Yellow	B11	-
6	02:10	Produce	Blue	R23	5 sec.
:	:	:	:	:	:

Table 3: Schedule structure

3.2 An Example

We present an example solution for the second problem proposed for the production of 6 batches of "*Blue*" and 6 batches of "*Green*" constrained by the deliver times of batches of "*Yellow*", "*Red*" and "*White*" given in table 4 and the due dates for an amount of 3 batches of both products given in Table 5.

The scheduling problem poses the following optimisation problem: A schedule which is a possible solution to the given delivery times of raw materials does not necessarily meet the given product due dates. The quality of a schedule then can be evaluated by the magnitude of the deviation from these due dates and optimisation algorithms can be used for optimising this objective.

White		Yellow	Red
0:00	4:20	0:00	0:10
0:30	5:30	0:30	0:40
2:00	6:20	1:10	1:50
2:20	6:55	4:20	5:30
3:00	7:10	5:40	6:50
3:35	8:00	6:45	7:50

Table 4: Deliver times in [min:sec] for the raw materials

3 bat. of Blue in B31	3 bat. of Green in B32
4:20	3:16
8:30	8:46

Table 5: Product due dates in [min:sec]

In Table 6, a schedule is given which has been derived from an optimisation using genetic algorithms [3]. For each product "*Blue*" and "*Green*" the table gives the six starting times in [min:sec] of the production of a batch of product and the corresponding reactor, in which the reaction shall take place.

Blue	Reactor	Green	Reactor
0:50	R23	1:32	R22
3:10	R23	2:22	R22
3:40	R23	2:33	R22
4:33	R23	5:30	R23
6:30	R23	7:25	R21
7:12	R22	8:05	R21

Table 6: Schedule consisting of the production times in [min:sec] and corresponding reactor

4 Control Architecture

This section describes the control architecture for the plant. It was developed in a ”top-down” oriented fashion, thus leading to a hierarchical structure. Although the control architecture was designed as it seemed to be the best to the authors for this problem, the resulting structure is quite typical for that of a chemical plant in general.

4.1 Control tasks and structure of the control program

Concerning the scheduling aspect of the plant, the main control task is to automatically run a schedule as described in Section 3.1. Another basic control task is safety monitoring and handling safety critical situations. Thus, apart from the automatic mode it also must be possible to operate the plant in other modes, e.g. in a *recovery* mode, if an alarm situation has been detected, or manually. For this an operating mode management is needed which switches between the different modes according to the user input and safety requirements. The main control tasks can be identified as

- Safety monitoring
- Operation mode management
- Automatic processing of a given schedule

These tasks are reflected in the hierarchical structure of the control program: The top layer consists of a function block "*Main*", calling a function "*Safety*" which checks for safety critical issues, e.g., if non of the liquid levels is above a critical threshold. It also checks whether the initial conditions for running a schedule, i.e. that all tanks are empty, no valve is open and all pumps are off, hold. Then a function block "*Mode*" for the determination of the operation mode according to user input variables and the results of the safety check is called. If the current operation mode is automatic operation or aborting automatic, a further function block "*Auto*" is called else the end of the main function block is reached. In Fig. 2 the structure of the top layer and the inputs and outputs of the function blocks called are illustrated in a function block language oriented manner. The function blocks "*Mode*" and "*Auto*" are described in the following Sections 4.2 and 4.3 in more

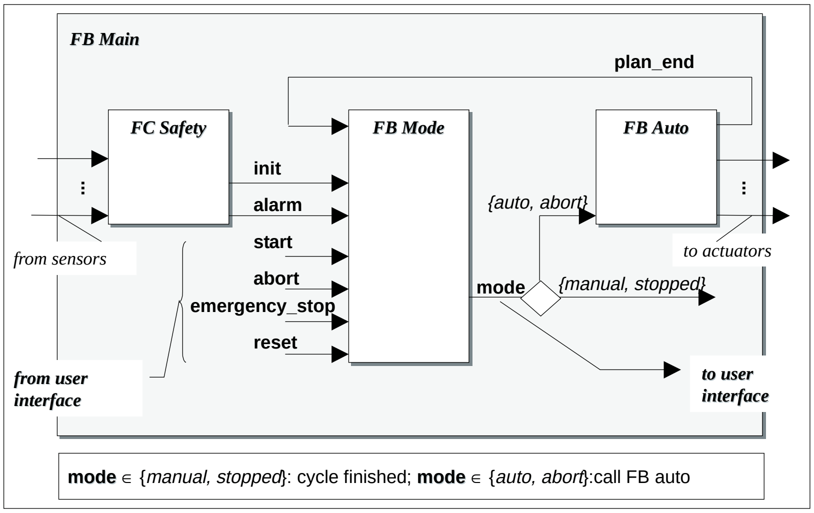

Figure 2: Top layer of the control program

detail. We chose this structure as such a hirarchical structure with a mode management on the top layer is quite typical for automation solutions. Also typical is the separation of safety and operation, with the safety monitoring as the first task of the cycle because it has the highest priority.

4.2 Operating mode management

The following operation modes shall be realized:

- **Manual mode** for operating the plant manually (via the user interface) e.g. in order to rinse it.
- **Automatic mode** for running a given schedule on the plant.
- **Aborting mode** to stop a started schedule but wait until the already running recipes have finished.
- **Recovery mode** or inching, if an alarm has been detected or the user has pressed the "Emergency Stop Button".

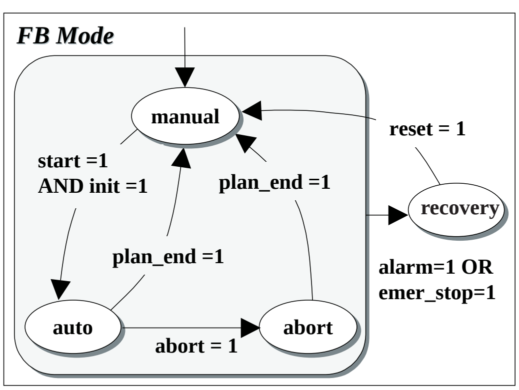

Figure 3: Mode management

The operation mode management e.g. the determination of the current mode can be specified as an automaton, which is shown in Fig. 3. The operating modes are represented by the states of the automaton, the transitions from one mode to another are given by logical operations between values of user input variables and output variables from e.g the safety check function.

4.3 Automatic mode - realizing a given schedule

To run a given schedule as described in Sec. 3 on the plant, the function block "*Auto*" is called, whose principal structure is shown in Fig. 4. "*Auto*" calls fur-

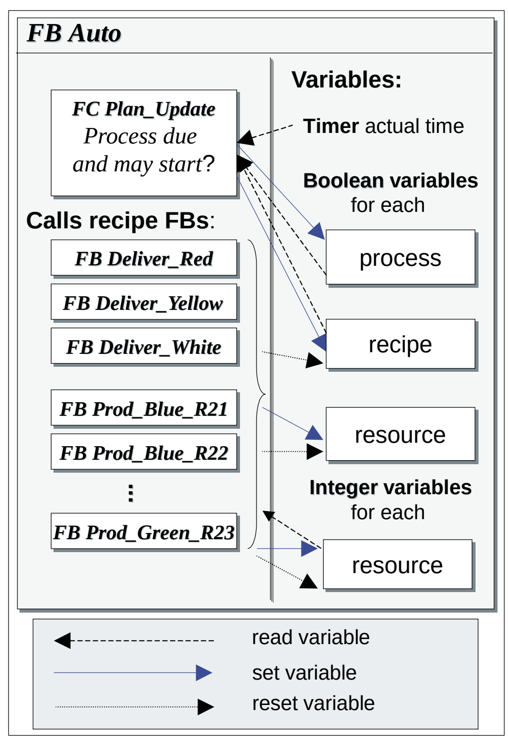

Figure 4: Structure of the automatic mode function block

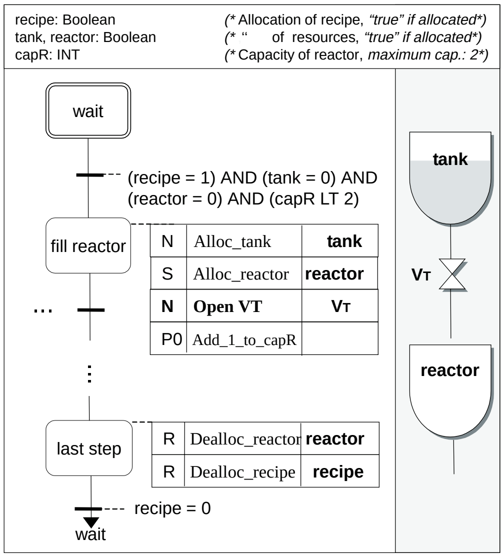

Figure 5: Structure of a recipe function block

ther function blocks, implemented as Sequential Function Charts (SFCs), each corresponding to a recipe as defined in section 3. The principle structure of such a recipe SFC is illustrated in Figure 5. As the task is not only to simply run a schedule on the plant, but also to consider that processes that are too late still are processed in the order which is given by the schedule, Boolean variables are introduced for all processes (i.e. entries of the schedule), all recipe function blocks and all shared resources (tanks, reactors). These variables are semaphores, which are set to denote that the respective process etc. is allocated and reset to denote that it is free, according to the following rules:

- A recipe is allocated to a process when the process is due and the recipe is available. The recipe deallocates itself when it terminates.

- A resource is allocated to a recipe when it is needed and available. It is deallocated by the recipe when no longer needed.

Additionally, integer variables are introduced to model the capacity of the resources (tanks, reactors), e.g. the maximum capacity of a product buffer tank is 3 as it may contain three batches. Fig. 4 illustrates which function blocks can set and which can reset these semaphores and capacity variables.

The scheduler then works as follows.

1. When changing from manual to automatic mode, e.g. when the automatic FB is called for the first time, a timer, indicating the "actual time" is started.

2. Then for every every process, the function block "*Plan_Update*" is called which checks the following:
   - The corresponding starting time of the process given in the schedule is compared to the "actual time".
   - If the process is due, it is checked whether the recipe needed is (still) allocated to another process.
   - If the required recipe is free (e.g. recipe function block in step "*idle*") the semaphore for the respective process and that for the recipe are set.

3. The recipe function blocks are called. The recipes do not immediately start but are initially in an idle step waiting until the following further conditions hold :
   - The recipe semaphore has been set.
   - The resources (tank and/or reactor) that are needed in the following step are not allocated to another process.
   - The resource which is to be filled has enough free capacity.

   When a recipe reaches the last step it resets its semaphore and goes back to the idle step waiting to be processed for an other process. Fig. 5 shows the principle structure of a recipe function block (given as Sequential Function Chart (SFC) according IEC1131-3 [2]) with the idle step as first step and the transition condition (given in Structured Text (ST)) which need to be satisfied before the recipe can start.

5 Conclusions

We presented a batch plant which is suited to serve as a case study for scheduling and control problems and a control architecture for this plant. In the specification phase of the design process of the control logic, control tasks were specified by simple automata which were straightforwardly translated into Instruction List (IL) or were specified by sequential function charts. In our future research, we intend to integrate simulation and verification techniques into the design process.

6 Acknowledgement

We would like to thank the European Community which sponsored this plant.

References

[1] http://www-verimag.imag.fr/VHS/.

[2] IEC1131-3. *Standard IEC 1131: Programmable controllers, Part 3: Programming Languages*. IEC Geneva, Switzerland, 1993.

[3] A. Liefeldt, T. Löhl, M. Stobbe, and S.Engell. Simulation and scheduling of recipe-driven batch processes based on a single data model. page 12, 1999.