SYSTEMATIC SYNTHESIS OF CLEANING PROCEDURES FOR PIPELINE NETWORKS USING PETRI NETS

Hung-Hsiang Chou Chuei-Tin Chang *
Li-Ping Chung

* *Department of Chemical Engineering, National Cheng Kung University, Tainan Taiwan 70101*

Abstract: Cleaning pipeline networks is a routine operation in almost every chemical plant. Our aim is to develop a systematic strategy to generate the needed operating procedures with Petri-net models. Specifically, the proper material-transfer routes are selected on the basis of the Petri-net representation of all possible paths. The equipment models are then attached to this path model according to network configuration to build a system model. By connecting the system model with a net representing the schedule manager, a recipe can be produced accordingly to achieve a multi-route sequential/concurrent schedule. The effectiveness of this approach is demonstrated with a realistic example.

Keywords: Petri net, reachability tree, pipeline network, cleaning operation.

1. INTRODUCTION

Cleaning pipelines is one of the routine operations that has to be performed in any chemical plant. The cleaning procedure can essentially be viewed as the operation steps to transfer a detergent, a disinfectant or an inert material from the inlets (sources) to the outlets (sinks) of a pipeline network. There have been a few related studies reported in the literature. All of them are concerned with the operating procedures for material transfer in pipeline networks. The most relevant work was published by Foulkes et al. (1988). They represented the states of fragments in a plant structure with a series of condition lists. A combination of artificial intelligence techniques, pattern matching and path search algorithms were adopted to identify all feasible routes for transferring a designated material from one storage tank to another in the plant. Uthgenannt (1996) used digraph models to describe the network of interconnected process equipments. The material transfer routes and the required operating procedures were obtained using a graph search method. Viswanathan et al. (1998) adopted a software called iTOPS to synthesize the operating procedures for batch plants. They used Grafchart (which is a modified version of Grafcet and Petri net) to model and represent operating procedures in a hierarchical way. The model elements were linked to a knowledge base developed within the G2 environment. Gabbar et al. (2004) handled this problem with the recipe formal definition language (RFDL). RFDL editor and parser were proposed to synthesize master recipe and the corresponding control recipe was also automatically generated accordingly.

It should be noted that the above-mentioned results are not directly applicable in the present application. First of all, the objective of a cleaning operation is in general not the same as that of simple material transfer. It is important in the former case to ensure that all parts in the pipeline network are included in the material transfer routes, while this constraint is not imposed in the latter. Secondly, it is difficult to generate valve-sequencing steps with the available methods to achieve a multi-route cleaning schedule. Thus, the objective of present study is mainly the development of a Petri-net based strategy to synthesize the operating procedures to perform multiple material-transfer tasks for cleaning the entire network. For the sake of brevity, an introduction of the basic features of Petri net is not provided here. The readers can refer to the general reviews given in Peterson (1981) and David et al. (1994).

2. REPRESENTATION OF MATERIAL-TRANSFER PATHS

The first critical issue in modelling any network should be concerned with the division of the system into distinct components. The concept of the piping *fragments* (Foulkes et al., 1988) is adopted in this work for this purpose. In particular, a fragment is defined as a collection of pipeline branches and/or processing units separated from other fragments by the valves, pumps and other means of flow blockage in the pipeline network. Let us consider Figure 1 as an example. Eight fragments can be identified according to this definition, i.e., $FR1$ - $FR8$. Notice that, in this case, every pump and its isolation valves are viewed as one *lumped* power-generating system and this system is treated as a flow blockage if it is turned off.

By connecting the Petri-net models of all fragments according to the network configuration, the *path model* in Figure 2 can be constructed. Here, the places $FR1$ - $FR8$ are used to reflect the fragment states. A token entering one of these places denotes the condition that detergent is delivered to the corresponding fragment from an upstream source fragment. Each of the places labelled with 'PK' is used to keep a record of the connection status between two neighboring fragments. A transition in this model can be considered as the operator/controller action to remove the corresponding flow blockage. Notice that, if a valve permits bi-directional material transfer, then two transitions should be adopted to characterize these two opposite flows.

Notice that, in the Petri net given in Figure 2, every record-keeping place is connected with *inhibitor arcs* to the input transitions of the place representing its downstream fragment. This is due to the need to impose additional constraints are to eliminate the possibilities of infinite material-transfer loops caused by the existence of bi-directional valves. With these inhibitors arcs, each fragment-representing place can be visited by a token only once. Without them, a token may travel endlessly in one of the following two loops: (1) $FR3 \rightarrow FR4 \rightarrow FR3 \rightarrow FR4 \cdots$ and (2) $FR5 \rightarrow FR6 \rightarrow FR5 \rightarrow FR6 \cdots$.

3. ENUMERATION OF POSSIBLE ROUTES

Since there can be more than one route emanating from a particular source fragment to the sink fragments of a pipeline network, it is desirable to first identify all of them to ensure thoroughness of the cleaning operation. This task can be achieved by constructing a *reachability tree* (Murata, 1989; Wang et al., 2002) from a given initial condition on the basis of the Petri-net model. Let us consider the Petri-net model presented in Figure 2. By assuming that the detergent is stored in tank $T1$ initially, the corresponding reachability tree (shown in Figure 3) can be generated and its markings can be found in Table 1. In order to conveniently identify the elements in a marking, the token numbers are classified into 2 subsets and arranged sequentially in a vector, i.e., $\mathbf{M}_k = [\mathbf{FR}_k | \mathbf{PK}_k]$, and $k = 0, 1, 2, \cdots, 10$. Here, each subset label is identical to the place labels of its elements. Based on this convention, useful information can be directly acquired from the marking of each terminal node in the reachability tree. Specifically, the sink fragment of a material-transfer route should be associated with 1 in the 1st subset $\mathbf{FR}_k$ of the terminal node. The corresponding connection status among various fragments in the network can be

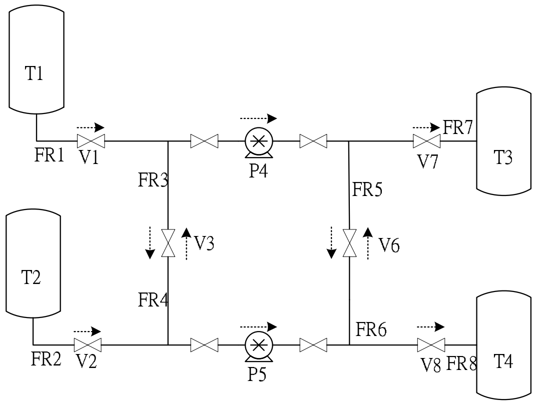

Fig. 1. An example of pipeline network

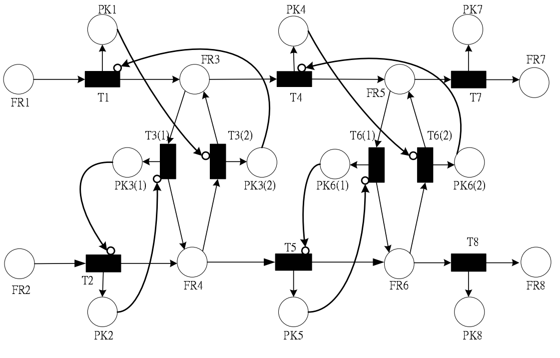

Fig. 2. The Petri net of pipeline network example

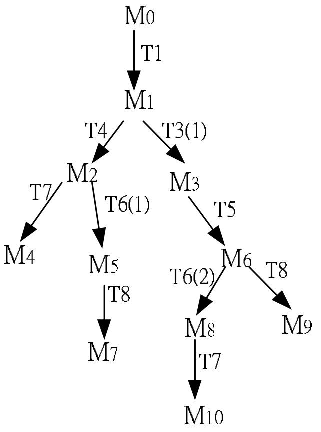

Fig. 3. A reachability tree of Fig. 2

Table 1. Markings $\mathbf{M}_k$s in Fig. 3.

k	$\mathbf{FR}_k$	$\mathbf{PK}_k$
0	{ 1 0 0 0 0 0 0 0 }	{ 0 0 0 0 0 0 0 0 0 0 }
1	{ 0 0 1 0 0 0 0 0 }	{ 1 0 0 0 0 0 0 0 0 0 }
2	{ 0 0 0 0 1 0 0 0 }	{ 1 0 0 0 1 0 0 0 0 0 }
3	{ 0 0 0 1 0 0 0 0 }	{ 1 0 1 0 0 0 0 0 0 0 }
4	{ 0 0 0 0 0 0 1 0 }	{ 1 0 0 0 1 0 0 0 1 0 }
5	{ 0 0 0 0 0 1 0 0 }	{ 1 0 0 0 1 0 1 0 0 0 }
6	{ 0 0 0 0 0 1 0 0 }	{ 1 0 1 0 0 1 0 0 0 0 }
7	{ 0 0 0 0 0 0 0 1 }	{ 1 0 0 0 1 0 1 0 0 1 }
8	{ 0 0 0 0 1 0 0 0 }	{ 1 0 1 0 0 1 0 1 0 0 }
9	{ 0 0 0 0 0 0 0 1 }	{ 1 0 1 0 0 1 0 0 0 1 }
10	{ 0 0 0 0 0 0 1 0 }	{ 1 0 1 0 0 1 0 1 1 0 }

identified from $\mathbf{PK}_k$. From the reachability tree given in Figure 4, it can be seen that there are four terminal nodes, i.e., $\mathbf{M}_4$, $\mathbf{M}_7$, $\mathbf{M}_9$ and $\mathbf{M}_{10}$. The fragments in these four material-transfer routes can be are shown in the first four rows of Table 2. Similarly, another reachability tree can be built from the second source fragment $FR2$ and four more material-transfer routes can be found accordingly. Their fragments are given in the last four rows of Table 2.

4. ROUTE SELECTION PROCEDURES

Although every material-transfer route identified from the reachability trees of a Petri-net model can be adopted to clean a portion of the given pipeline network, it may not be necessary to include all of them to achieve the operation objective. In this study, the task of cleaning a pipeline network is considered to be accomplished if the detergent is transported through every fragment at least once. In addition, there are strong incentives to identify non-overlapping routes so that the concurrent cleaning strategies can be devised accordingly. For illustration convenience, let us consider all possible cleaning routes of the pipeline network given in Figure 1, i.e., routes (1) - (8). These routes can be arranged as a *path matrix* shown in Table 2. The specific steps of our route selection procedure are presented in the sequel:

Table 2. Path Matrix.

Route No.	$FR1$	$FR2$	$FR3$	$FR4$	$FR5$	$FR6$	$FR7$	$FR8$
1	○		○		○		○	
2	○		○		○	○		○
3	○		○	○	○	○	○	
4	○		○	○		○		○
5		○		○		○		○
6		○		○	○	○	○	
7		○	○	○	○	○		○
8		○	○	○	○		○	

(1) Select a row in the path matrix with the largest number of elements.
(2) Identify and temporarily remove another row with at least one element in the column where the row selected previously in step 1 also has an element. Repeat this step until no more rows can be identified.
(3) Repeat steps 1 and 2 until all rows are exhausted.
(4) Recover the temporarily removed rows.
(5) Delete all the rows selected in steps 1 - 3.
(6) Delete all columns in which the elements of the rows selected in steps 1 - 3 are located.
(7) Repeat the above steps until all columns are deleted.

Notice that the above steps may have to be repeated several times before termination. Since the routes selected each time in steps 1 to 3 cannot be overlapping, it is possible to execute the material-transfer operations along these routes concurrently. However, the routes selected in separate batches must be cleaned sequentially. As an example, one can easily verify that applying the proposed procedure to Table 2 leads to a operation policy of cleaning routes (3) and (5) in sequence.

5. EQUIPMENT MODELS

In order to generate the specific operation steps to realize the cleaning tasks of the selected material-transfer routes, the Petri-net model of a pipeline network must contain not only the component models of fragments but also those of the installed equipments, such as the valves, pumps and compressors, etc. The valve model is presented in Figure 4. Here, the places PV denote two opposite valve positions respectively. The transitions TV represent the valve-switching actions. Notice that the place PC can be interpreted as the valve-switching requirement. The places PR can be used to record the actual times that the corresponding valve-switching actions have been carried out. Since it is possible to call for a material-transfer action when the corresponding valve is already open, transition $TR(O)$ is introduced as the output of $PC(O)$. A normal arc is adopted in the former case to avoid a token being permanently kept in place $PC(O)$, while a test arc is used in

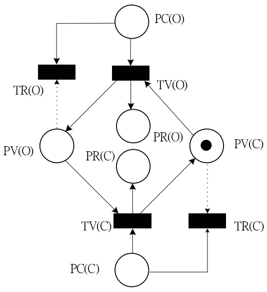

Fig. 4. Valve model.

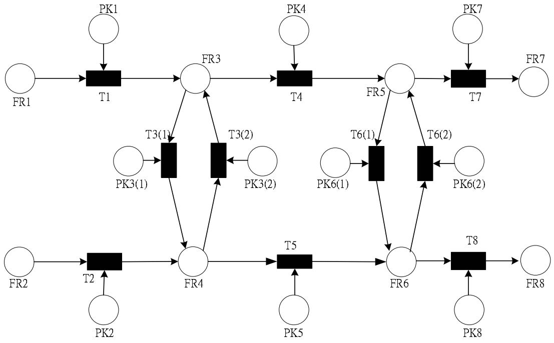

Fig. 5. Modified path model.

the latter to prevent loss of the tokens in $PV(O)$. Finally, note that the transition $TR(C)$ is adopted for the same reason.

Since the operating procedures of pumps, compressors and their isolation valves can be considered as standard industrial practices, e.g., see Karassik and McGuire (1998), their detailed steps are not described in the equipment models for the sake of simplicity. Specifically, the Petri net presented in Figure 4 is also used to model a power-generating system. In this case, the places $PV(O)$ and $PV(C)$ represent two opposite states, i.e., on and off, of the system respectively. The transitions $TV(O)$ and $TV(C)$ can be regarded as a series of standard operation actions to turn on and off the pump/compressor system.

6. GENERATION OF OPERATION STEPS

The operating procedure for cleaning a selected route in the pipeline network can be obtained based on a system model. To create such a model, the equipment models should be attached to a modified Petri-net representation of the material-transfer paths. This modified path model can be transformed from its original version by removing all inhibitor arcs and then reversing the directions of all connecting arcs between the record-keeping places and their input transitions. These places can now be interpreted as the demands to establish the corresponding connections between two neighboring fragments. Let us again consider the system in Figure 1 as an example. The path model in Figure 2 can be converted to the modified Petri net presented in Figure 5. This net is then expanded by connecting its transitions respectively to the places $PC(O)$s in the corresponding equipment models with normal arcs. A detailed listing of these connections can be found in the second column of Table 3. This practice is meant to reflect the relationship between each material-transfer action and the need to open the corresponding valve (or turn on the corresponding pump/compressor).

Table 3. Connections Between Path Model and Equipment Models.

Transition	Equipment($PC(O)$)	Equipments ($PC(C)$)
$T1$	$V1$	$V3$, $P4$
$T2$	$V2$	$V3$, $P5$
$T3(1)$	$V3$	$V2$, $P5$
$T3(2)$	$V3$	$V1$, $P4$
$T4$	$P4$	$V6$, $V7$
$T5$	$P5$	$V6$, $V8$
$T6(1)$	$V6$	$P5$, $V8$
$T6(2)$	$V6$	$P4$, $V7$
$T7$	$V7$	-
$T8$	$V8$	-

In order to guarantee the feasibility and safety of the material-transfer steps through a selected route, it is necessary to impose additional auxiliary control rules in operating the related valves, pumps and/or compressors. These *equipment operation rules* are summarized below:

Equipment Operation Rules: *Given a specific material-transfer action, all valves and/or pumps surrounding its downstream fragment (except the one used for facilitating the present material transfer) should be closed/switched off.*

To realize these rules, additional normal arcs should be introduced to connect the transition representing the given action in the modified path model to the places $PC(C)$s in the Petri-net models of the equipments surrounding its downstream fragment. In the case of our example system in Figure 1, these additional connections are shown in the third column of Table 3. As a result, a cleaning procedure can be correctly generated from *any* given initial condition by executing the system model. To be specific, let us assume that the valves $V3$, $V6$ and $V7$ are left open after cleaning route (3), while the other valves are closed and both pumps are off. The operating procedure to clean route (5) in this situation can be found to be: close $V3$ and $V6$, open $V2$ and $V8$, and then turn on $P5$.

7. EXECUTION OF MULTIPLE TASKS

In this study, it is assumed that two cleaning tasks can be scheduled sequentially or concurrently according to the two Gantt charts shown in Figures 6 respectively. It is required in the former case that $t_0 < t_1 \leq t_2 < t_3$, while in the latter case, the constraint is either $t_0 \leq t_1 < t_2 \leq t_3$ or $t_0 \leq t_1 < t_3 \leq t_2$. The time needed to accomplished a particular task should be determined on a case-by-case basis. The systematic approach for generating the operation steps of a single-route cleaning task has already been presented in the previous sections. Here, the proposed Petri-net based techniques are extended to synthesize the operating procedure for executing a multi-task schedule. In order to coordinate the implementation times of various different tasks according to the given schedule, the Petri-net model of a *schedule manager* is attached to the system model. If the tasks of cleaning routes (3) and (5) in the example system are to be carried out according to the first schedule given in Figure 6, this schedule manager can be modelled with the Petri net given in Figure 7.

Notice first that, instead of a single event, transition $TX1$ is now interpreted as a collection of operation steps (events) to establish route (3) for cleaning purpose and, similarly, $TX6$ represents another set of operation steps to establish route (5). To activate the operation steps corresponding to route (3), transition $TX1$ is connected to the place representing source fragment, i.e., $FR1$, and also to those representing the operation demands for establishing fragment connections, i.e., $PK1$, $PK3(1)$, $PK5$, $PK6(2)$ and $PK7$. Similarly, $TX6$ is connected to $FR2$, $PK2$, $PK5$ and $PK8$ to trigger the operation actions for cleaning route (5). On the other hand, transitions $TX4$ and $TX9$ represent the actions to terminate the cleaning operations of routes (3) and (5) respectively. It is assumed in the present study that each termination procedure consists only two steps, i.e., switching off the pump and then closing the exit valve of source fragment. The connections from $TX4$ and $TX9$ to the places in system model reflect the demands for these actions. Notice also that the delay times of $TX1$, $TX4$, $TX6$ and $TX9$ are assigned to meet the given schedule exactly. Another type of connections between the Petri-net model of the schedule manager and the system model are concerned with places $P2$, $P4$, $P6$ and $P9$. They are used simply for maintenance purposes. Places $P2$ and $P6$ mark the initialization phases of the operations to establish routes (3) and (5) respectively, while $P4$ and $P9$ represent the preparation stages prior to the termination steps for these two routes respectively. It should be noted that every such place and all the places labelled with '$PR(O)$' and '$PR(C)$' in system model are connected to a common output transition. Since the operation records of pumps and valves are stored in the latter places, this practice is in essence to reset these records before carrying out each of the above four distinct sets of operation steps.

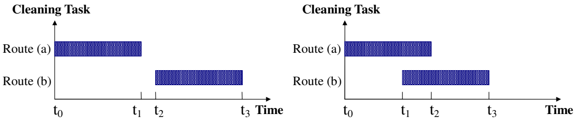

Fig. 6. Gantt charts of sequential and concurrent schedules.

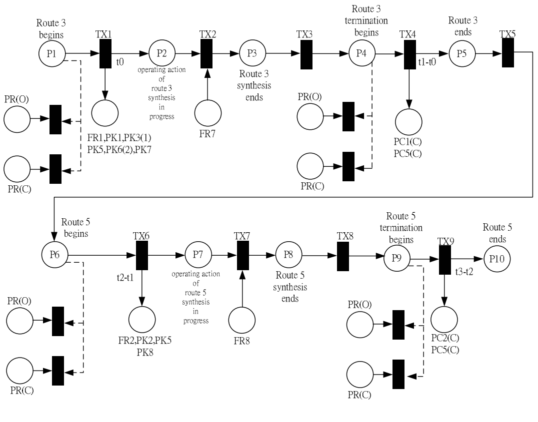

Fig. 7. The Petri net of schedule manager.

Table 4. Operation Steps for Cleaning the Pipeline Network in Fig. 1

Time	Operation Steps
t_0	Open valves $V1$, $V3$, $V6$ and $V7$. Switch on pump $P5$.
t_1	Switch off pump $P5$. Close valve $V1$.
t_2	Close valves $V3$ and $V6$. Open valves $V2$ and $V8$. Switch on pump $P5$.
t_3	Switch off pump $P5$. Close valve $V2$.

For illustration convenience, let us assume that all valves are closed and all pumps are off initially in the example system. To realize the schedule in Figure 6, an operating procedure can be generated by placing a token in the place $P1$ and then executing the simulation run. The resulting operation steps are presented in Table 4.

8. CASE STUDY

The pipeline network described in Foulkes et al.(1988) is adopted here as a realistic example to demonstrate the capability of the proposed method. The network contains 8 storage tanks, 36 valves and 4 pumps (see Figure 8). A total of 5 source fragments, i.e., $FR1$ - $FR5$, 20 internal fragments, i.e., $FR6$ - $FR25$, and 6 sink fragments, i.e., $FR26(1)$ - $FR28(1)$ and $FR26(2)$ - $FR28(2)$, can be defined in this system. It is assumed that there are no upper limits imposed upon the amounts of detergent stored in the source tanks, i.e., $T1$ - $T5$, and also the capacities of the sink tanks, i.e., $M1$ - $M3$. In addition, the spent material gathered from separate cleaning routes are allowed to be stored in the same sink tank. Since each sink tank has two inlets, they are thus treated as two distinct fragments in this example. Finally, it is also assumed that all valves are closed and all pumps are turned off initially.

The cleaning routes can be selected on the basis of the path model of the given pipeline network (see Table 5). It can be observed that the required cleaning tasks must be implemented *sequentially* in three stages. During each stage, multiple material-transfer operations can be executed *concurrently* via the selected routes. For illustration convenience, let us further assume that the operation periods needed to carried out the different concurrent operations in the same stage are identical. Specifically, the operation periods of the three stages are assumed to be $[t_0, t_1]$, $[t_2, t_3]$ and $[t_4, t_5]$ respectively. The complete operating procedures to achieve the corresponding schedules can be generated from the Petri-net based simulation runs. The results can be found in Table 6.

9. CONCLUSION

A systematic strategy is presented in this paper for generating detailed operating procedures to clean any given pipeline network. The cleaning routes are selected on the basis of the Petri-net representation of all material-transfer paths. The operation steps for transporting material through a designated route can be identified from the simulation results obtained with the Petri-net model of entire system. By connecting this net with one that represents a schedule manager, the multi-route cleaning recipes can also be produced with the proposed simulation techniques. The effectiveness of this approach is clearly demonstrated with a realistic example.

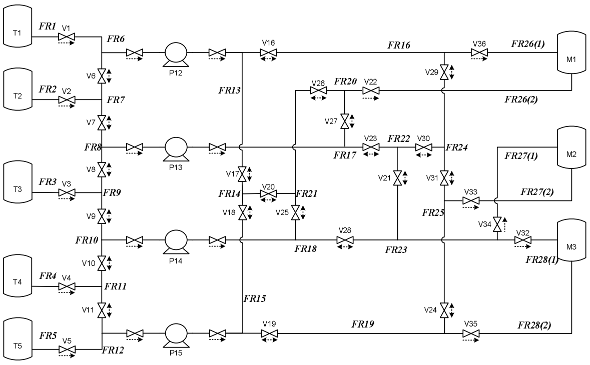

Fig. 8. A realistic Example of pipeline network.

Table 5. Cleaning Routes in Fig. 8.

Stage	Route(s)
1	• $FR1 \rightarrow FR6 \rightarrow FR7 \rightarrow FR8 \rightarrow FR9 \rightarrow FR10$ $\rightarrow FR11 \rightarrow FR12 \rightarrow FR15 \rightarrow FR19$ $\rightarrow FR25 \rightarrow FR24 \rightarrow FR16 \rightarrow FR13$ $\rightarrow FR14 \rightarrow FR21 \rightarrow FR18 \rightarrow FR23$ $\rightarrow FR22 \rightarrow FR17 \rightarrow FR20 \rightarrow FR26(2)$
2	• $FR2 \rightarrow FR7 \rightarrow FR6 \rightarrow FR13 \rightarrow FR16 \rightarrow FR26(1)$ • $FR3 \rightarrow FR9 \rightarrow FR10 \rightarrow FR18 \rightarrow FR23 \rightarrow FR28(1)$ • $FR4 \rightarrow FR11 \rightarrow FR12 \rightarrow FR15 \rightarrow FR19 \rightarrow FR28(2)$
3	• $FR5 \rightarrow FR12 \rightarrow FR15 \rightarrow FR19 \rightarrow FR25 \rightarrow FR27(2)$ • $FR1 \rightarrow FR6 \rightarrow FR7 \rightarrow FR8 \rightarrow FR17 \rightarrow FR22$ $\rightarrow FR23 \rightarrow FR27(1)$

Table 6. Operation Steps for Table 5.

Time	Operation Steps
t_0	Open valves $V1$, $V6$, $V7$, $V8$, $V9$, $V10$, $V11$, $V16$, $V17$, $V19$, $V20$, $V21$, $V22$, $V23$, $V24$, $V25$, $V27$, $V28$, $V29$ and $V31$. Switch on pump $P15$.
t_1	Switch off pump $P15$. Close valve $V1$.
t_2	Close valves $V7$, $V8$, $V10$, $V17$, $V21$, $V24$,$V25$ and $V29$. Open valves $V2$, $V3$, $V4$, $V32$, $V35$ and $V36$. Switch on pumps $P12$, $P14$ and $P15$.
t_3	Switch off pumps $P12$, $P14$ and $P15$. Close valves $V2$, $V3$ and $V4$.
t_4	Close valves $V11$, $V27$, $V28$, $V31$ and $V35$. Open valves $V1$, $V5$, $V7$, $V21$, $V24$, $V33$ and $V34$. Switch on pumps $P13$ and $P15$.
t_5	Switch off pumps $P13$ and $P15$. Close valves $V1$ and $V5$.

REFERENCES

David, R. and H. Alla (1994). Petri nets for modeling of dynamic systems - a survey. *Automatica* **30**:2, 175.

Foulkes, N. R., M. J. Walton, P. K. Andow and M. Galluzzo (1988). Computer-aided synthesis of complex pump and valve operations. *Comput. Chem. Eng.* **12**:9/10, 1035.

Gabbar, H. A., A. Aoyama and Y. Naka (2004). Recipe formal definition language for operating procedures synthesis. *Comput. Chem. Eng.* **28**, 1809-1822.

Karassik, I. J. and J. T. McGuire (1998). *Centrifugal Pumps*. 2nd ed.. pp. 885-887. Chapman & Hall. New York.

Murata, T. (1989). Petri nets: properties, analysis and applications. *Proc. IEEE* **77**:4, 541.

Peterson, J. L. (1981). *Petri Net Theory and the Modeling of Systems*; Prentice-Hall: Englewood Cliffs, NJ.

Uthgenannt, J. A. (1996). Path and equipment allocation for multiple, concurrent processes on networked process plant units. *Comput. Chem. Eng.* **20**:9, 1081.

Viswanathan, S., C. Johnsson, R. Srinivasan, V. Venkatasubramanian and K. E. Arzen (1998). Automating operating procedure synthesis for batch processes: Part I. Knowledge representation and planning framework. *Comput. Chem. Eng.* **22**, 1673-1685.

Wang, Y. F., J. Y. Wu and C. T. Chang (2002). Automatic hazard analysis of batch operations with Petri nets. *Reliab. Eng. Syst. Saf.* **76**, 91-104.