Downloaded from orbit.dtu.dk on: May 08, 2024

DTU Library

Synthesis of Valve and Pump Operations in Complex Plants by Using Functional Modeling

Song, Mengchu; Gofuku, Akio; Lind, Morten

Published in:
IFAC-PapersOnLine

Link to article, DOI:
10.1016/j.ifacol.2019.12.094

Publication date:
2019

Document Version
Publisher's PDF, also known as Version of record

Link back to DTU Orbit

Citation (APA):
Song, M., Gofuku, A., & Lind, M. (2019). Synthesis of Valve and Pump Operations in Complex Plants by Using Functional Modeling. *IFAC-PapersOnLine*, *52*(19), 187-192. https://doi.org/10.1016/j.ifacol.2019.12.094

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

Available online at www.sciencedirect.com

ScienceDirect

IFAC PapersOnLine 52-19 (2019) 187–192

Synthesis of Valve and Pump Operations in Complex Plants by Using Functional Modeling ⋆

Mengchu Song * **Akio Gofuku** ** **Morten Lind** ***

* *Graduate School of Natural Science and Technology, Okayama University, 700-8530 Okayama, Japan (e-mail: song.m.mif@s.okayama-u.ac.jp)*

** *Graduate School of Interdisciplinary Science and Engineering in Health Systems, Okayama University, 700-8530 Okayama, Japan (e-mail: gofuku-a@okayama-u.ac.jp)*

*** *Department of Electrical Engineering, Technical University of Denmark, 2800 Kongens Lyngby, Denmark (e-mail: mli@elektro.dtu.dk)*

Abstract: It is crucial to provide operators supports for operation planning in complex process plants. This paper presents an operation searching approach used for synthesis of valve and pump operations to establish specific pipe routes. Instead of modeling pipeline fragments, a functional modeling methodology called Multilevel Flow Modeling (MFM) is adopted to represent plant knowledge. Relying on causality feature of MFM, one can identify required state changes of specific functions in the model, and accordingly operations. Since relationships between functions, components, and operations are independent to the modeling object, these generic principles can be implemented into a rule-based reasoning system for inferring operational conditions of pipe routes, namely, operations. An example in the literature has been used to demonstrate the presented approach and application of the software.

© 2019, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.

Keywords: Operation planning, operating procedure synthesis, functional modeling, multilevel flow modeling, process industry.

1. INTRODUCTION

In the operation of complex process plants such as nuclear or chemical plants, it always needs operators to plan component operations to establish plant configurations that can achieve some goals. Because plant situations cannot be completely predicted, it is not realistic to prepare operation solutions for all requirements and constraints in advance. Due to complexity and unpredictability, it is necessary for operators to have external support such as relying on computer techniques for the planning activity.

Artificial Intelligence (AI) has been recognized as a solution for real world problems such as operation planning support. There are generally two aspects involved in the AI application: knowledge representation and inference. (Aylett et al., 2000) At first, the plant knowledge can be modeled in different formats such as state graphs (Ivanov et al., 1980), action template (Rivas and Rudd, 1975), and system ontology (O'Shima, 1978), which are used to develop knowledge-based systems for operation planning.

Operation planning is naturally formulated as a searching problem. Operations on specific components should be found out to achieve a required plant configuration. Since there are generally enormous numbers of components, mostly valves and pumps of pipework, there can be many operation combinations and accordingly many pipe routes, as referred to the situation of combination explosion (Rivas and Rudd, 1974). In other words, the searching space would be huge even though most of the combinations are not likely for the current plant situation. Therefore, it needs good heuristics rather than blind searching to find a small group of alternative operation solutions from the large searching space. Different modeling methodologies plant knowledge can have respective advantages to provide a proper heuristic for searching component operations.

Foulkes et al. (1988) presented a work of searching possible pipe routes to establish charging paths in a blending system. The plant knowledge is modeled by means of "fragments" of pipelines. Prolog (Programming in Logic) language is adopted as the searching approach because it is suitable for problems involving logic, pattern matching and searching. Then, requirements of operations on valves and pumps for establishing pipe routes are searched using a forward strategy. Although modeling plant knowledge as pipeline fragments is simple, it is still a massive and loss-prone work to model all the fragments.

This paper presents an operation synthesis approach based on a graphical functional modeling called Multilevel FLow Modeling (MFM) (Lind, 2011) to solve the same problem in Foulkes et al. (1988). MFM is capable to model inten-

⋆ This work is supported by Japan Society for the Promotion of Science(JSPS) [KAKENHI grant number 16H03136].

2405-8963 © 2019, IFAC (International Federation of Automatic Control) Hosting by Elsevier Ltd. All rights reserved.
Peer review under responsibility of International Federation of Automatic Control.
10.1016/j.ifacol.2019.12.094

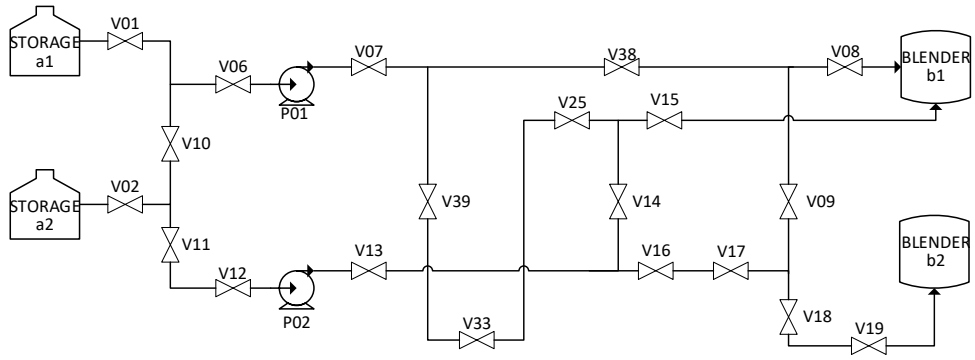


Fig. 1. The schematic of a blending system

tional knowledge of process plants in terms of system goals and functions. Not only the physical level of pipework, the other abstraction levels of plants such as abstract functions and system objectives can be represented. Moreover, the causal aspect of MFM can explain cause-consequence relations between components represented by functions, which are therefore potential for making causal reasoning, such as fault diagnosis and action planning (Lind and Zhang, 2014; Song and Gofuku, 2018).

The paper is constructed as follows: Section 2 presents the problem statement for operation searching and synthesis. Section 3 briefly introduce the theory of MFM. How to search operations on valves and pumps based on MFM is illustrated in Section 4. Section 5 presents the application of a rule-based reasoning system for synthesizing operations. Finally, conclusions are drawn in Section 6.

2. PROBLEM STATEMENT

To simplify demonstration, the system used in this paper is extracted from the original case, as shown in Fig. 1. It is a blending system used for blending materials from different sources that includes 2 storage tanks, 2 blenders, 2 pumps and 20 valves. The initial problem was to develop a software that can produce one or multiple pipe routes to achieve an operational goal of charging materials from a storage tank to a blender such as:

$$Transport\ fluid\ materials\ from\ a1\ to\ b1.$$

Pipe routes are those descriptions of desired states of specific components in the system, which thus can be also expressed as requirements of component operations.

2.1 Modeling "Fragments"

Foulkes et al. (1988) modeled the plant knowledge by representing "fragments" of pipeline via which materials can flow through. For instance, the actual flow in the pipeline from tank $a1$ to the center of valve $v01$, namely $j01$ can be modeled as:

$$link\ a1\ to\ j01\ by\ [v01(op)]. \tag{1}$$

The square bracket indicates conditions that are required for the statement "*link* $a1$ to $j01$", in the above fragment there is just one condition, i.e. $v01$ open, which means that materials can flow from $a1$ to $j01$ only if valve $v01$ is open. Some pipeline fragments can have several conditions including state requirements on multiple components, the following is the pipeline from the center of valve $v10$ ($j10$) to the center of valve $v06$ ($j06$),

$$link\ j10\ to\ j06\ by\ [v01(cl),\ v10(op),\ v06(op)]. \tag{2}$$

Where op and cl represent the valve state of open and close, respectively. ru emerged in the following expressions of pipeline fragments can represent the running state of pumps. Note that the flow via valve $v10$ can have two possible direction, either from valve $v10$ to valve $v06$ or valve $v01$ to valve $v10$. Apart from the fragment (2), hence, the following fragment that includes valve $v10$ also exists,

$$link\ j01\ to\ j10\ by\ [v01(op),\ v10(op),\ v06(cl)]. \tag{3}$$

The pipeline containing pump can be described as,

$$link\ j06\ to\ j07\ by\ [v06(op),\ p01(ru),\ v07(op)]. \tag{4}$$

2.2 Searching for Paths

A program should be designed to collect fragments until a flow path from a specific start to a specific end can be found. For instance, a set of fragments as following can be identified to establish a flow path from $a1$ to $b1$:

$$\begin{aligned}
&link\ a1\ to\ j01\ by\ [v01(op)],\\
&link\ j01\ to\ j06\ by\ [v01(op),\ v10(cl),\ v06(op)],\\
&link\ j06\ to\ j07\ by\ [v06(op),\ p01(ru),\ v07(op)],\\
&link\ j07\ to\ j38\ by\ [v07(op),\ v38(op),\ v39(cl)],\\
&link\ j38\ to\ j08\ by\ [v38(op),\ v08(op),\ v09(cl)],\\
&link\ j08\ to\ b1\ by\ [v08(op)].
\end{aligned}$$

Conditions in the square bracket of each identified fragment can be collected as the conditions for the found flow path. These conditions also indicate the necessary operations on valves and pumps on this identified path. In above case, therefore, the required operations are:

$v01,\ v06,\ v07,\ v38,\ v08$	*open*
$v10,\ v39,\ v09$	*close*
$p01$	*start*

3. MULTILEVEL FLOW MODELING

Instead of describing the plant topology by pipeline fragments, in this paper, a graphical functional modeling called Multilevel Flow Modeling (MFM) (Lind, 2011) is adopted for modeling plant knowledge.

3.1 Modeling Language

In MFM, plant's intentional knowledge, such as functions and goals are decomposed and integrated along means-end and whole-part dimensions. MFM has its own modeling language, i.e. graphical symbols, as shown in Fig. 2. There are six basic functional primitives being able to describe functions that involve flows of materials, energy, and information. Relations are used to describe dependencies between functions.

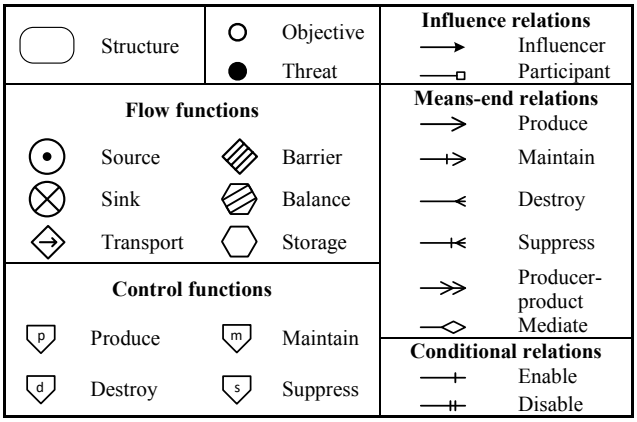


Fig. 2. Basic MFM symbols

3.2 Causality Feature

Besides representing plant knowledge, MFM is also able to reason about plant performance, majorly causes and consequences relationships. Once there is an evidence or assumption of a state change occurred in a function, the influence can be propagated to the neighboring functions along different consequence paths, as shown in Fig. 3. Meanwhile, this evidence or assumption can also result from the neighboring functions. Therefore, there can be several paths that can be tracked back along to know about where the influences come from. Note that above causal aspect of MFM exists in not only the whole-part dimension, but also the means-end dimensions.

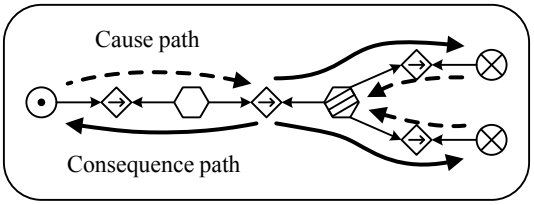


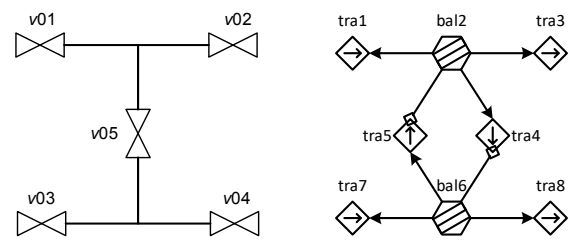
Fig. 3. Reasoning about causes and consequences in MFM

Descriptions of functions by MFM are more intuitive and effective way for modeling pipelines like those expressed as "link A to B". While the capability of reasoning about causes provides a potential approach for searching establishment conditions, i.e. how functions change their states in potential pipe routes. Because functions in MFM are corresponded to components, the conditions on functions are able to provide requirements for components, i.e. operations. In Section 4, the strategy for searching operations based on MFM will be introduced.

4. OPERATION SEARCHING STRATEGY

4.1 The MFM model

Modeling of two-way pipeline Fig. 4 shows a strategy for modeling a common pipework connection, i.e. two-way pipelines, in which materials can flow along either two opposite directions. Two *transport* functions in opposite directions are modeled between two *balance* functions representing pipe branches. Because only one *transport* function can be valid at one time, the modeling of influence relations between *transport* functions and *balance* functions should ensure that even existence of *transport* function in another direction will not propagate any influence on *transport* function in the current direction. The modeling strategy is to use *participant* relation at the downstream of *transport* function and *influencer* relation is used at the upstream. In the definition of MFM, *participant* means that the functions pointed by the relation cannot affect the current function.



(a) Two-way pipeline (b) Modeling by MFM

Fig. 4. Modeling of two-way pipeline by MFM

The actuator role In MFM, roles can be defined to describe how functions are related to the structural entities (Lind, 2010). If a function state can be directly achieved by an operation on the corresponding component, an *actuator* role can be defined for this function to describe a feature of controllability for the function. The category of the *actuator* role is defined according to the category of component. As shown in Fig. 5, the *actuator* role is graphically designed as a circle with α in the middle. In this paper, two *actuator* roles, i.e. *valve-operation* (*vao*) and *pump-operation* (*puo*) are defined for the *transport* function to indicate that there can be an operation on valve and pump, respectively. The symbol of the *actuator* role can be used for generating specific operations, which will be explained in the following.

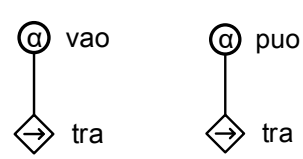


Fig. 5. Graphical definition of the *actuator* role in MFM

The whole MFM model of the blending system is shown in Fig. 6, which will be later used as the knowledge base for the operation searching. The model is basically constructed according to the abstraction level of pipework distribution. Two-way pipelines are modeled with above modeling strategy. There is only one mass flow structure, in which two *source* functions represent two storage tanks,

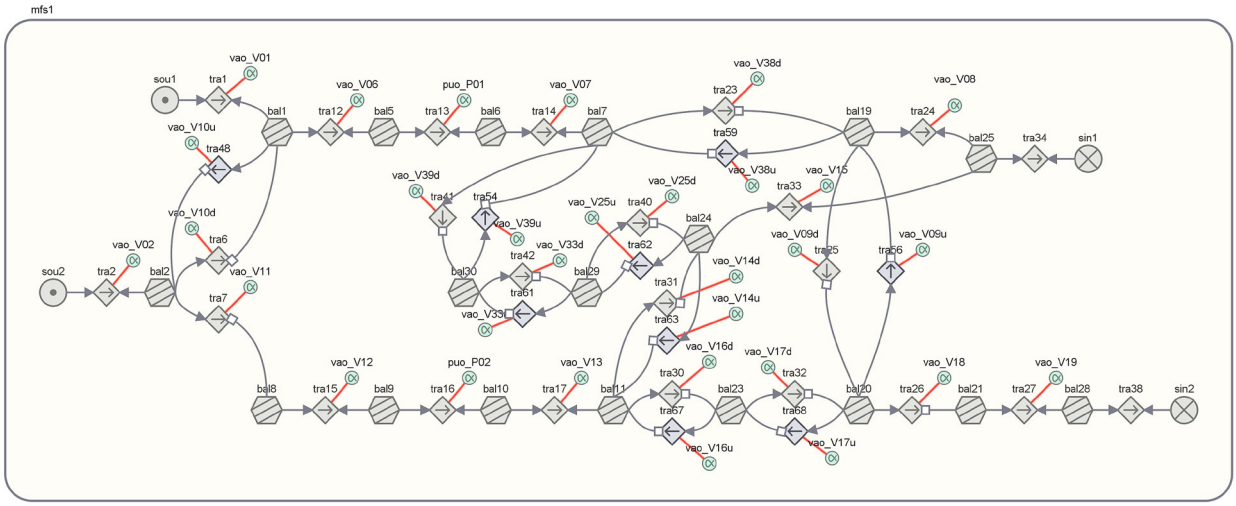


Fig. 6. The MFM model of the blender system

and two *sink* functions describe two blenders. Flow transfers in the system are represented by *transport* function regardless of what component it represents. As for pumps and valves, the *transport* function is used to describe the flow from a piece of pipeline in the upstream of component to that in the downstream. *Balance* function is used to describe pipe branches and the other connections between transport functions. *transport* functions that can be controlled by components are attached with *actuator* roles.

4.2 A rule-based operation searching strategy

In terms of MFM, the goal of synthesizing operations for establishing charging pipe routes could be to make the *sink* function representing the prescribed blender in a high state. Therefore, the operation searching should start from setting an assumption of a high state in the *sink* function. The causality feature of MFM will be used for reasoning about conditions, i.e. function states for possible pipe routes. The identified conditions will provide requirements for operations. There are two categories of generic reasoning rules that can be adopted for searching desired contents in the MFM model.

Causal reasoning rules In MFM, there exists dependency relations between different function connections, which reflect general disciplines of physical phenomenons and thus are independent to the specific modeling object. All the dependency relations in MFM have been implemented as rules for reasoning about causes and consequences of abnormal states in process plants. (Zhang et al., 2013). Table 1 shows the typical causal reasoning rules of MFM that may be used in the case of this paper. Because the system is modeled with only one flow function structure, only parts of causal reasoning rules for MFM patterns connected by influence relations are used. Full causal reasoning rules can be found in Zhang et al. (2013). Note that there are many MFM patterns in the case involved in *transport-balance* connections, hence combinations of rules of No.(3-5) in Table 1 can be used for continuous causal reasoning.

Operation identifying rules Assumptions in functions will provide requirements of component states, and hence

Table 1. Typical causal reasoning rules

No.	MFM pattern	Assumption	Conclusion (cause)
1	tra1 ← sin1	sin1 high	tra1 high
2	sou1 → tra1	tra1 high	sou1 high
3	tra1 ← bal1 → tra2	tra2 high/bal1 normal	tra1 high
4	tra1 ← bal1 → tra2	tra1 low/bal1 normal	n/a
5	tra1 ← bal1 → tra3; tra2 → bal1	tra3 high/bal1 normal	tra2 high tra1 low

operations on corresponding components. (Song and Gofuku, 2018) Table 2 lists these general reasoning rules used for deriving operations. Similar with states of MFM functions, the *actuator* role, i.e. *vao* and *puo* can have their own states, i.e. operations. For instance of a rule, high or low state of the *transport* function representing a valve will require the *actuator* role *vao* in the state of open or close for the achievement of each function state.

Table 2. Rules for identifying operations

No.	MFM pattern	Assumption	Conclusion (operation)
1	vao1 → tra1	tra1 high	vao1 open
		tra1 low	vao1 close
2	puo1 → tra1	tra1 high	puo1 start
		tra1 low	puo1 stop

4.3 Operation Searching Process

Before searching operations, it is assumed that there has been a charging path in the plant only it is unknown which the path it is. The aim here is to identify the cause-consequence relations behind this possible path. With the knowledge base constructed based on MFM, thus, the process of searching operations for establishing pipe routes would be:

Step 1 Determine the *sink* function that represents the prescribed blender expected to be charged. Assume a high state of this *sink* function.
Step 2 Identify a state of a neighboring function of the initial *sink* function that can result in the goal state by matching a generic rule of MFM.
Step 3 If there is an *actuator* role that is attached to the identified function and an operation identifying rule can be matched, then generate the operation. If not, go to Step 4.
Step 4 If the function state in Step 2 is a high state of the *source* function that describes the prescribed source of charging paths, i.e. storage tank, then a pipe route has been found, go to Step 5. Otherwise, set this function state as the new assumption. Go to Step 2.
Step 5 Present results.

The process for searching operations is essentially similar to the rule-based method presented by Zhang et al. (2013). More rules are implemented for the special requirements. The causal reasoning rules are used for reasoning about conditions for possible pipe routes, and the operation identifying rules are used for reasoning about the operation realization for pipe routes.

5. APPLICATION

5.1 The Rule-based Reasoning System

A rule-based reasoning system has been developed by using Drools for realizing the operation searching process based on MFM. Fig. 7 shows the algorithm behind the system. The core of the system is a rule engine, which implements an optimized Rete Algorithm called ReteOO. (Kumar, 2011) The rule engine can match working memory (or facts) and production memory (or rules) to infer conclusions. Situations that multiple rules are matched can be solved by the conflict resolution strategy to select the most proper rules. Firing rules will result in actions, which may generate new facts and update the rule base. There is also truth maintenance that can be used to avoid duplicated or conflicted results of inference.

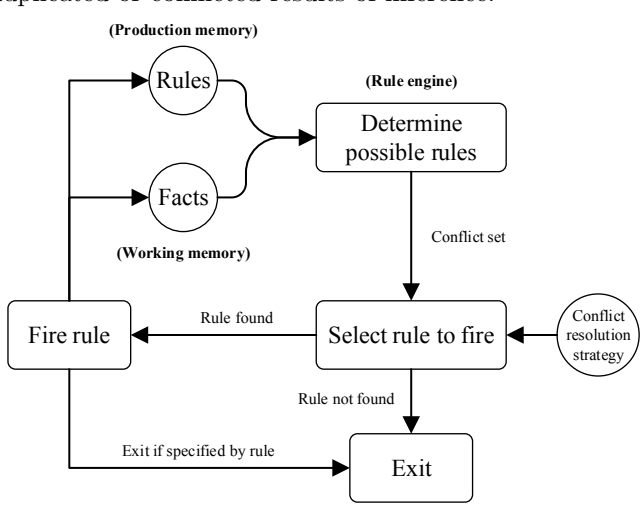


Fig. 7. The mechanism of the rule-based reasoning system

During searching operations, the first assumption on function, namely high state of the *sink* function can be set and stored in the working memory. One causal reasoning rule can be matched and fired to generate a new assumption on function, which will be stored in the working memory as well. Meanwhile, operation identifying rule might also be matched to generate an assumption on *actuator*, i.e. operation. The new assumption on function will result in firing the other rules until no rule can be matched. All inferred results that are stored in the working memory can be output as a tree structure.

5.2 The results

The rule-based reasoning system is applied into the case presented in the Section 2. In order to generate pipe routes realizing the original goal, i.e. *transport fluid materials from* $a1$ *to* $b1$, we set the consequence of establishment of possible pipe routes as the first assumption, i.e. [$sin1$, high] and input it into the reasoning system. The results generated by the system is a tree structure rooted by the first assumption. All the other assumptions derived from this root node are listed in the branches. The whole result for the case is not shown here. A piece of the generated tree structure shown in Fig. 8 is used for demonstrating different categories of nodes in the result.

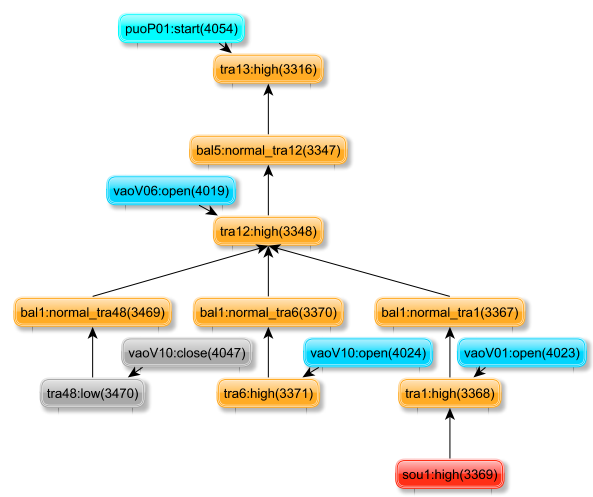


Fig. 8. Demonstration of node colors in the inference result

Green and blue nodes mean that operation identifying rules can be matched to generate operations on pumps and valves, respectively, which are placed on above of the corresponding assumptions on functions. Grey nodes are low state of *transport* functions derived from high state of *transport* functions, and accordingly operation of closing. In reality, they are invalid to contribute to a pipe route, because we cannot expect to establish a charging by closing valves. The operation of closing valve is only valid for isolating alternative paths, which has not been discussed at the moment. Red nodes are inferred assumptions on *sink* functions, which means that no rules can be further matched to generate inference. Orange nodes are the other assumptions of function states.

Different pipe routes that can achieve the goal can be identified from the tree structure. Each path starting from $sou1$ *high* in the terminal branch, ending with $sin1$ *high* on the root can indicate a valid charging pipe. The

assumptions on the *actuator* role attached to the identified paths, therefore, are the requirements of operations on corresponding valves and pumps. Note that there must be a pump operation to provide pressure resource for each identified pipe route. In summary, there are eight pipe routes, accordingly eight operation solutions of for achieving the original goal, as follows:

(1) Open $v01$, open $v06$, start $p01$, open $v07$, open $v39$, open $v33$, open $v25$, open $v15$;
(2) Open $v01$, open $v10$, open $v11$, open $v12$, start $p02$, open $v13$, open $v16$, open $v17$, open $v09$, open $v38$, open $v39$, open $v33$, open $v25$, open $v15$;
(3) Open $v01$, open $v10$, open $v11$, open $v12$, start $p02$, open $v13$, open $v14$, open $v15$;
(4) Open $v01$, open $v06$, start $p01$, open $v07$, open $v38$, open $v09$, open $v17$, open $v16$, open $v14$, open $v15$;
(5) Open $v01$, open $v06$, start $p01$, open $v07$, open $v38$, open $v08$;
(6) Open $v01$, open $v10$, open $v11$, open $v12$, start $p02$, open $v13$, open $v14$, open $v25$, open $v33$, open $v39$, open $v38$, open $v08$;
(7) Open $v01$, open $v06$, start $p01$, open $v07$, open $v39$, open $v33$, open $v25$, open $v14$, open $v16$, open $v17$, open $v09$, open $v08$;
(8) Open $v01$, open $v10$, open $v11$, open $v12$, start $p02$, open $v13$, open $v16$, open $v17$, open $v09$, open $v08$.

Apparently, (5) is the "shortest" path, that is, the pipe route that needs least equipment items, or operations, which is one of the criteria proposed in Foulkes et al. (1988).

It is also important to generate alternative paths for one goal in case of component failures in abnormal situations. With the presented reasoning system, it could be flexible to switch paths when some paths are not available due to component failures. In the future, the capacity of the reasoning system on planning alternative pipelines should be tested with some failure scenarios, in which parts of components may not be able to work.

6. CONCLUSION

Although operations of opening valves and starting pumps for establishing pipe routes of material charging can be inferred by applying the rule-based reasoning system, it is difficult to generate "off" operations, such as closing valves and stopping pumps for isolating a pipe route from all alternatives since pipe routes are found by searching influences of opening valves that can establish flows, how the influences of closing valves to isolate flows is not considered. But it is clear that at least neighboring valves of each identified pipe routes should be closed. In addition, the issue of synthesizing operation sequences after identification of pipe routes has not been discussed in the paper. The other operating constraints concerned with valves and pumps, as well as safety conditions should be further considered for an appropriate operation sequence.

MFM has advantages of describing multiple abstraction levels of plant's intentional knowledge. The MFM model in this paper only shows an abstraction level as same as what presented in the pipeline configuration. From a larger scale of plant, systems may have different intentions and be interrelated by means and end. Thus, it is unrealistic to automatically generate an MFM model from plant configuration. In turn, the planning should not consider only one abstraction level of plant's means-end structure. Necessary actions for solving a specific problem are expected to be searched from different systems as long as relevant knowledge is represented by MFM.

ACKNOWLEDGEMENTS

This work is supported by Japan Society for the Promotion of Science(JSPS) [KAKENHI grant number 16H03136]. The first author would also like to thank Technical University of Denmark (DTU) for providing opportunity of external research stay.

REFERENCES

Aylett, R., Petley, G., Chung, P., Chen, B., and Edwards, D. (2000). AI planning: Solutions for real world problems. *Knowledge-Based Systems*, 13(2), 61–69.

Foulkes, N.R., Walton, M.J., Andow, P.K., and Galluzzo, M. (1988). Computer-aided synthesis of complex pump and valve operations. *Computers and Chemical Engineering*, 12(9-10), 1035–1044.

Ivanov, V.A., Kafarov, V.V., Perov, V.L., and Reznichenko, A.A. (1980). On algoeithmization of the start-up of chemical productions. *Engineering cybernetics*, 18(1), 104–110.

Kumar, N. (2011). Rule based programming with Drools. *International Journal of Computer Science and Information Technologies*, 2(3), 1121–1126.

Lind, M. and Zhang, X. (2014). Functional modelling for fault diagnosis and its application for NPP. *Nuclear Engineering and Technology*, 46(6), 753–772.

Lind, M. (2010). Knowledge Representation for Integrated Plant Operation and Maintenance. In *7th Nuclear Plant Instrumentation, Control, and Human-Machine Interface Technologies (NPIC&HMIT)*, 2092–2103. American Nuclear Society, Las Vegas, NV, USA.

Lind, M. (2011). An introduction to multilevel flow modeling. *Nuclear safety and simulation*, 2(1), 22–32.

O'Shima, E. (1978). Safety supervision of valve operations. *Journal of Chemical Engineering of Japan*, 11(5), 390–395.

Rivas, J.R. and Rudd, D.F. (1974). Synthesis of Failure-Safe Operations. *AIChE Journal*, 20(2), 320–325.

Rivas, J.R. and Rudd, D.F. (1975). Man-Machine Synthesis of Disaster-Resistant Operations. *Operations Research*, 23(1), 2–21.

Song, M. and Gofuku, A. (2018). Planning of alternative countermeasures for a station blackout at a boiling water reactor using multilevel flow modeling. *Nuclear Engineering and Technology*, 50(4), 542–552.

Zhang, X., Lind, M., and Ravn, O. (2013). Consequence reasoning in multilevel flow modelling. *IFAC Proceedings Volumes (IFAC-PapersOnline)*, 12(PART 1), 187–194.