1. Phlip decision support case

This document introduces a real-world case scenario that describes the need for decision support in information gathering. We use this example case to show for a different case what is the feasibility of our approach and its benefits. This case is inspired based on [1] and in particular we used appendix A as an inspiration for the CMMN user model.

1.1. Problem description

Phlip produces medical scanners. As part of a contract Phlip also offers to service and maintain those scanners during a certain period. The scanners are very complex and require a lot of expertise to be maintained. One of the expertises that is needed is the ability to diagnose a cause for a failure to dispatch the right maintenance engineer. Phlip uses a process consisting of remote, on-site and expert options to diagnose the failure. These options are crucial to collect the necessary information for a correct diagnosis. Currently, Phlip uses a fixed order of subsequent process steps to collect information. However, it would like to switch to a more dynamic model where the outcome of earlier options determines what next option to take.

The process of diagnosing a failure works as follows. Phlip receives a notification of a failing scanner from a client. An engineer receives this notification and commences the process of finding the cause for this failure. After performing an internal DIP in which different options can be used to collect the right information, Phlip makes a certain diagnosis. In this case we assume a scanner to consist of five subsystems of which one or multiple must be diagnosed as the cause. The subsystems are:

- Software
- Table Hardware
- Control systems
- Magnetic system
- Gradient system

The engineer has a set of field service engineers (FSE) available that can repair this machine. But, for every failed subsystem that the FSE does not have the capability

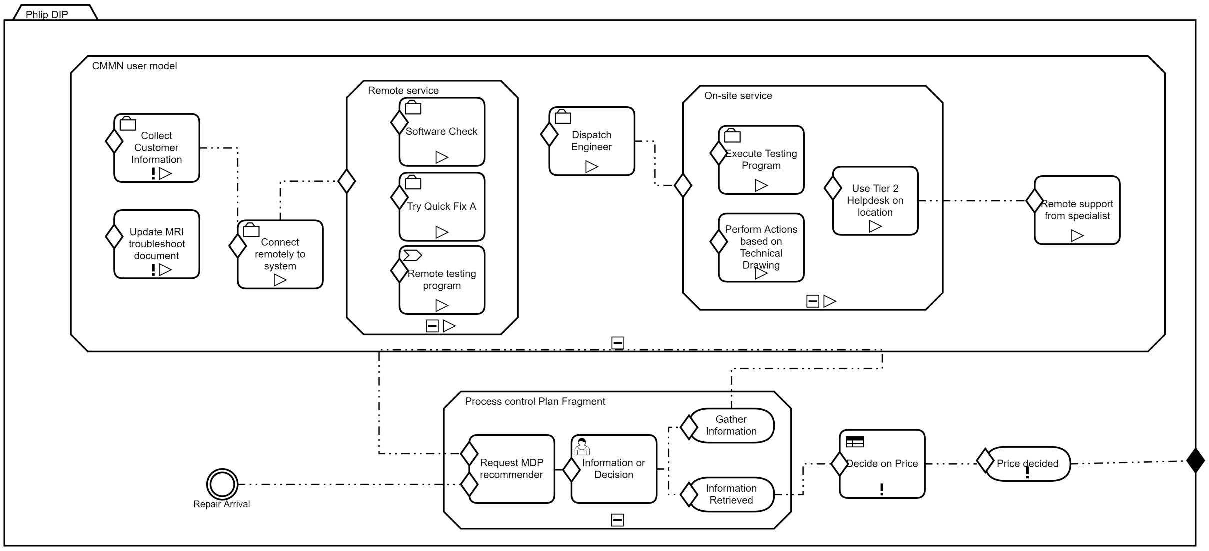

Figure 1: CMMN plan for Phlip

	Case File items	Distribution values	Distribution
A.	Software Check	{[-5,0,0,0,0],[3,0,0,0,0],[1,0,0,0,0],[2,0,0,0,0]}	$[\frac{1}{8}, \frac{3}{8}, \frac{3}{8}, \frac{1}{8}]$
B.	Try Quick fix A	{[0,0,0,0,10],[0,0,0,10,0],[0,10,0,0,0],[0,0,10,0,0]}	Binomial(0.2)
C.	Remote Testing program	{[1,0,1,0,0],[0,1,0,1,1],[1,1,1,1,1],[3,0,3,0,3]}	$[\frac{1}{3}, \frac{1}{6}, \frac{1}{6}, \frac{1}{3}]$
D.	Execute Test Program	{[0,0,-3,-3,-3],[0,0,3,3,3]}	Bernoulli(0.3)
E.	Perform Actions based on drawings	{[2,0,0,2,0],[0,3,1,0,1],[0,0,0,-2,0],[0,0,0,3,0]}	Binomial(0.5)
F.	Use tier 2 helpdesk	{[3,0,0,0,0],[0,3,0,0,0],[0,0,3,0,0],[0,0,0,3,0]}	Binomial(0.2)
G.	Remote support from specialist	{[0,-4,0,4,0],[0,0,4,-4,0],[5,-2,-2,-2,-2],[0,0,0,0,10]}	$[\frac{1}{6}, \frac{1}{3}, \frac{1}{3}, \frac{1}{6}]$

Table 1: Case File items for Koffer case

to repair, an extra penalty cost of 10000 is incurred for delay and extra work. Phlip is looking for an approach to support the decision makers by informing them about the value of gathering more information through other options and the value of currently choosing which FSE to dispatch. Figure 1 shows how the CMMN user model looks.

In this case, we focus on the internal DIP. At the moment that the notification arrives, Phlip has no exact information concerning the properties of the scanner that is failing. Koffer can gather information for this scanner through the data attributes listed in Table 1.

1.2. Design-time approach

In this section we discuss the necessary steps in our approach to optimize the DIP of Phlips.

Retrieve MDP structure. Every research action that is taken improves the knowledge about the true underlying failure diagnosis. To determine the subsystems that have actually failed, we use a threshold. If any of these five subsystems have an indicator value that is higher than four, we assume this subsystem is failing. The final decision that this process should deliver is which service engineer to dispatch. There is a set of four service engineers available, each with a different set of capabilities. Dispatching is costly which is why there is always one engineer sent.

$$\mathcal{F} = \{(1,1,0,1,0),(0,1,0,1,1),(1,0,1,0,1),(0,1,1,1,0)\}$$

To retrieve the MDP structure of the example DIP we need an information structure for the decision-making task. We define the following information structure, based on the case file items introduced in Table 1 and the Koffer case,

$$Y(\mathbf{s}, f) = \sum_{x \in \text{Subsystems}} 10000 \cdot (1 - f_x)\mathbb{1}(\sum_{s \in \mathbf{s}} s_x > 4)$$

Where f_x is the x-th value in the vector f, and s_x is the x-th value of variable s. Phlip aims to optimize the process of gathering information through the information of these attributes and hence to dynamically increase the certainty about which subsystem(s) can be diagnosed as the cause of the failure. This increased certainty improves the effectiveness of the diagnostics process. At the same time, Phlip loses efficiency because of precious time that is lost while gathering information. Hence, they need a DIP to optimally make the trade-off between time and information. Figure 1 shows the CMMN plan for the internal DIP. This figure contains more tasks than the information-acquiring tasks of Table 2, since the entire DIP includes other tasks as well.

Filter MDP model. For this case, we do not discuss the details of the MDP. To see how the process and data that has been defined are rewritten to states and actions we refer to the main paper.

	Stages	Cost	Preceding	Required	
1.	Phlip DIP	0	()	0	
α.	Perform Remote Service	5	(4)	0	
β.	Perform On-Site Service	0	()	0	
	Procedural tasks	Cost	Preceding	Required	
2.	Collect customer information	30	()	1	
3.	Update scanner troubleshoot document	30	()	1	
4.	Connect remotely to system	20	(2)	0	
5.	Dispatch Diagnosing Engineer	400	(2,3)	0	
6.	Dispatch Tier 3 Helpdesk	600	(12)	0	
	Information acquiring tasks	Cost	Preceding	Required	Case File Item
7.	Software Check	200	(α)	0	A
8.	Try Quick fix A	60	(α)	0	B
9.	Remote Testing program	100	(α)	0	C
10.	Execute Test Program	300	(β)	0	D
11.	Perform Actions based on drawings	300	(β)	0	E
12.	Use tier 2 helpdesk	500	(β, 10)	0	F
13.	Remote support from specialist	400	(6)	0	G

Table 2: Procedural tasks for Koffer case

2. Deployment

We can now deploy the solution in the CMMN environment. For a randomly drawn state we will discuss what is currently available information and how the decision maker is assisted in making the best decision.

Table 3 shows an example of an information and plan state, while Table 4 shows the possible actions that can be taken in this state. Currently dispatching the tier 3 helpdesk is not yet allowed because first the tier 2 helpdesk must have been asked. Therefore, we can only collect information variables E or F. However, there is no constraint on when to finish the process, which means that we can also take a final decision in this state. It appears that it is even optimal to stop searching for further information and to decide to dispatch the first field service engineer for this job.

3. Numerical analysis

Based on the case description from [1], we compare the optimal solution to a solution as they are often used in practice. This solution follows the natural order that is also described in [1]. But as soon as we are in the situation that one FSE perfectly fits the state of the diagnosis, we decide to dispatch this FSE directly. Finally, if all information

Table 3: Current state

		Information state
A	Software Check	[3,0,0,0,0]
B	Try Quick fix A	[0,0,0,10,0]
C	Remote Testing program	[0,1,0,1,1]
D	Execute Test program	[0, 0, -3, -3, -3]
E	Perform Actions based on drawings	⊥
F	Use tier 2 helpdesk	⊥
G	Remote support from specialist	⊥

Plan state	
Software Check	1
Try Quick fix A	1
Remote testing program	1
Execute test program	1
Perform actions based on drawings	0
Use tier 2 helpdesk	0
Remote support from specialist	0
Collect Customer Information	1
Update scanner troubleshoot document	1
Connect remotely to system	1
Dispatch diagnosing engineer	1
Dispatch tier 3 helpdesk	0
Perform remote service	1
Perform on-site service	1

Table 4: Action set

Possible tasks/stages	**Expected cost**
Perform Actions based on drawings	1976.5
Use tier 2 helpdesk	1805.69

Final Decisions	**Expected cost**
(1,1,0,1,0)	**1786.66**
(0,1,0,1,1)	7273.33
(1,0,1,0,1)	7480.00
(0,1,1,1,0)	8819.99

has been collected, we can maximize the profit by calculating which FSE is optimal to dispatch. Table 5 shows how these two solutions perform. The MDP clearly has the lowest cost in expectation. It pays a higher cost for collecting information, but is able to use that information in a better way.

Table 6 shows nine randomly drawn states that have been tested for both solutions. What we notice is that the alternative solution has a lot of cases where it collected all information before deciding on a solution. The MDP solution however has found ways to determine the optimal solution without always collecting all the information. Note that in case 9 the MDP is not better than the alternative. For this case the MDP has been too quick with deciding and the alternative has found a better decision by collecting all information. This again shows the value of an approach where next algorithms, human decision makers can still be involved to optimize decisions.

Similar to the first case in the main paper, we can conclude that the method is succesful and able to decrease the cost compared to an alternative method that follows the normal flow of this DIP. We would like to state that the way of modeling this problem

Table 5: Comparison MDP vs Decision Tree

Information	*MDP*	*Alternative*
Expected cost	2938.0	4580.0
Number of final states	263	953
Average retrieval cost	770.18	541.38

Instance	*Final Information State [A,..,G]*	*MDP*	*Alternative*
1.	[[3,0,0,0,0],[0,0,0,0,10], [3,0,3,0,3], [0, 0, -3, -3, -3], [2,0,0,2,0], [0, 0, 3, 0, 0], [0, -4, 0, 4, 0]]	145	2645
2.	[[1,0,0,0,0], [0,0,0,10,0], [1,0,1,0,0], [0, 0, 3, 3, 3],[0,3,1,0,1], [0, 0, 3, 0, 0], [0, -4, 0, 4, 0]]	1445	2645
3.	[[3,0,0,0,0], [0,0,0,10,0], [3,0,3,0,3], [0, 0, -3, -3, -3], [0,3,1,0,1], [0, 0, 3, 0, 0],[0, -4, 0, 4, 0]]	2445	2645
4.	[[3,0,0,0,0], [0,0,0,10,0], [0,1,0,1,1], [0, 0, -3, -3, -3],[0,3,1,0,1], [3, 0, 0, 0, 0],[5, -2, -2, -2, -2]]	445	2645
5.	[[-5,0,0,0,0], [0,0,0,10,0], [3,0,3,0,3], [0, 0, -3, -3, -3], [2,0,0,2,0], [0, 0, 3, 0, 0], [0, -4, 0, 4, 0]]	1245	2645
6.	[[3,0,0,0,0], [0,0,0,10,0], [1,1,1,1,1], [0, 0, -3, -3, -3], [0,3,1,0,1], [0, 0, 0, 3, 0],[5, -2, -2, -2, -2]]	2145	2645
7.	[[2,0,0,0,0], [0,10,0,0,0], [0,1,0,1,1],[0, 0, 3, 3, 3],[0,3,1,0,1],[3, 0, 0, 0, 0],[5, -2, -2, -2, -2]]	2645	10445
8.	[[-5,0,0,0,0], [0,0,10,0,0], [0,1,0,1,1], [0, 0, 3, 3, 3], [0,3,1,0,1], [3, 0, 0, 0, 0], [0, -4, 0, 4, 0]]	10145	12645
9.	[[-5,0,0,0,0], [0,0,0,10,0], [1,1,1,1,1], [0, 0, -3, -3, -3], [0,0,0,3,0], [3, 0, 0, 0, 0], [0, 0, 0, 0, 10]]	11445	2645

Table 6: Comparison of cost for 9 instances

is complicated. In the real world, there could be examples where there is directly a very clear diagnosis to make. In that case it would be highly unlikely that further diagnostic activities could still lead to different conclusions. For future work with these DIPs we still want to improve the way of modeling the DIP also to get even closer to reality.

References

[1] A. Shahrestani, Data-Driven Corrective Maintenance: MR Root Cause Analysis from Machine Logs, Master's thesis, Eindhoven University of Technology, `https://pure.tue.nl/ws/portalfiles/portal/146442643/Master_Thesis_Arash_Shahrestani.pdf` (2020).