

# Engineering the Decision-Making Process Using Multiple Markov Theories and DEMO

Sérgio Guerreiro<sup>(✉)</sup>

Lusófona University, Campo Grande 376, 1749-024 Lisbon, Portugal  
`sergio.guerreiro@ulusofona.pt`

**Abstract.** In the current fast-changing and turbulent operational environments, the organizations are continually being pressured by many endogenous and exogenous environmental variables. Many and complex effects occur simultaneously and large volumes of data are available. For this reason, in a process-based organization, when change is demanded (*e.g.*, business processes re-engineering) it is difficult to collect, and interpret, the complete information about the current state of the organization. Therefore, a problem is how to decide which design actions should be enacted with the incomplete information available from the executed business processes. In this context, this paper combines information systems engineering (DEMO business transactions design) and operation research (Markov theories) to contribute to the decision-making body of knowledge. As the result, this solution enforces the organization with resiliency capabilities that are triggered whenever any misalignment occurs. The proposed solution is evaluated through argumentation and by a qualitative comparison between two Markov theories (MDP and POMDP) based on a real-world case study.

**Keywords:** Decision-making · Management · MDP · Observation · POMDP · State · Value

## 1 Introduction

Decision-making is a management competence [16,20] that encompasses: the intelligence to discover the organizational problems, the design of potential solutions, the choosing of the best solution, the implementation of the solution and the verification if the new solution fulfills the desired goals. These stages occur in many levels of organizational management, *e.g.*, project management, operational management, middle management, *etc.*

On the one hand, multiple endogenous and exogenous factors promote the need to enforce a continuous decision-making process, for instance, requirements change, legal changes or fraud attempts. In response to these multiple changes, it is necessary to have native decision-making capabilities that continuously find innovative solutions to adapt the organizational operation to be more efficient

and effective. In this context, the study of mechanisms to engineer the informed decision-making [24] are key competence for the success of the organization's management.

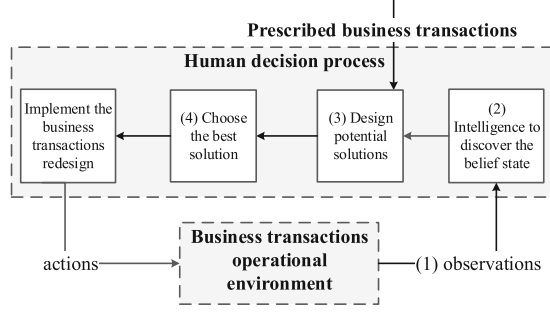
On the other hand, the business processes play a dual role: *(i)* they are the result of applying design constraints for a particular organizational reality [15], and are valid over a given period of time, and *(ii)* operational support to the actions performed by actors, by other words, business process guide actors in acting. The actors have an active and autonomous role in the execution of business processes, therefore, it is not guaranteed that the requirements of business processes are met properly on the daily routines. For example, if a company's recommendation is to always obtain a written record when contacts are made to the clients, nothing limits the ability of an actor to contact a client directly, by phone, without leaving any trace of the communication. The same example can be applied to the financial markets, with a huge adverse impact potential to the organization and to its environment.

In this context, combining decision-making with business processes, under complex process-based environments, raise the following challenges *(i)* inability to map the current operational observations with the current state where the organization actually is [17], *e.g.*, when actors perform workarounds [1] and override the previous defined prescriptions then the manager need to collect more information to interpret what, in fact, was executed; and *(ii)* incomplete observations [3], *e.g.*, because its too expensive to collect information, or, if the business processes are partially performed in paper by humans and partially machine-based. Therefore, in the majority of the situations, the management should support their decisions in partial information about the surrounding environment (also named as partial observable environments).

In light of this, in this paper, we narrow the decision-making management problem to the business transactions operation optimization. So forth, we propose and evaluate an innovative approach combining DEMO-based business process design [5,7]) and operations research (using Markov decision process (MDP) and partial observable Markov decision process (POMDP) theories [19]). DEMO obliges the full specification of business transaction dynamics and MDP and POMDP yields the greatest amount of utility over some number of decision steps.

Figure 1 provides an overview of our approach. The steering cycle of observation (cf. Fig. 1(1)), assessing the environment (cf. Fig. 1(2)), designing the potential solutions (cf. Fig. 1(3)) and choosing the best solution (cf. Fig. 1(4)). These steps recall to the management competences and we emphasize that they are mainly human based. Nevertheless, we argue and show how automatic tools deliver support to the managers, aiding at some point in their decision-making tasks.

The rest of the paper follows a simplified design science research (DSR) approach [14], encompassing the iterations of: problem statement, design of a solution for the given problem and evaluation phase. Firstly, Sect. 2 identifies the problem statement boundary and the background concepts (MDP, POMDP and DEMO) that are needed in the rest of the paper. Then, in Sect. 3, the design



**Fig. 1.** Overview of our approach.

for an informed decision-making process is detailed. After that, Sect. 4 is devoted to the explanation of a case study that was previously introduced in [11]. Next, from the preceding results, Sect. 5 evaluates the solution for the given case study, using argumentation and a qualitative comparison between MDP and POMDP when applied to the context of business transactions. Afterwards, Sect. 6 presents the conclusions and future work.

## 2 Background Concepts

### 2.1 Markov Theories

In probabilities theory, a Markov process is a stochastic process that satisfies the Markov property [19]: if the transition probabilities from any given state depend only on the actual state and not on previous history. By other words, the predictions for the future are solely based on its present state. Its future and past are independent. Markov theories are applied to systems that are controlled or uncontrolled (autonomous) *versus* observable or partial observable. Where, a system is completely observable if every state variable of the system affects some of the outputs. And, a process is said to be completely controllable if every state variable of the process can be controlled to reach a certain objective in finite time by some unconstrained control action.

A Markov chain is used to refer to a process which has a countable and discrete set of state spaces, yet not controllable. When the states of the process are only partial observable, then an hidden Markov model (HMM) should be used. From this point forward, to engineer the decision-making process, we narrow our research in the controllable systems.

A Markov decision process (MDP) is able to solve the problem of calculating an optimal policy in an accessible and stochastic environment with a known transition model [18]. A MDP is defined by the tuple  $(S, A, T, R, \gamma)$ .

In partial accessible environments, or whenever the observation does not provide enough information to determine the states or the associated transition probabilities, then the hidden Markov model (HMM) or partially observable Markov

decision process (POMDP) solutions should be considered. The difference is that HMM is applied to uncontrolled systems and POMDP to controlled systems. A POMDP solution provides a rich framework for planning under uncertainty [11]. A POMDP finds a mapping between observations (not states) to actions. In practice, two different states could appear to be observed equally. A POMDP is defined by the tuple  $(S, A, Z, T, O, R, \gamma)$ .

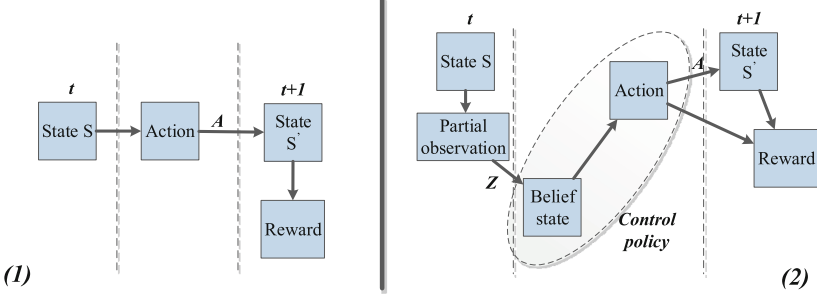
The definitions for the MDP and POMDP tuples are:  $S$  is a set of states, representing all the possible underlying states the process can be in, even if state is not directly observable;  $A$  is a set of actions, representing all the available control choices at each point in time;  $Z$  is a set of observations, consisting of all possible observations that the process can emit;  $T : A \times S \times S \rightarrow \prod(S)$  is a state transition function, where  $\prod(\cdot)$  is a probability distribution over some finite set, encoding the uncertainty in the system state evolution;  $O : A \times S \times Z \rightarrow \prod(Z)$  is an observation function, relating the observations to the underlying state;  $R : A \times S \times S \times Z \rightarrow \mathbf{R}$  is an immediate reward function, giving the immediate utility for performing an action of the underlying states;  $\gamma$ : discounted factor of future rewards, meaning the decay that a given achieved state suffers through out time.

For a POMDP, at each period, the environment is in some state  $s \in S$ . The manager takes an action  $a \in A$ , which causes the environment to transition to state  $S'$  with probability  $T(S'|S, a)$ . And because the manager does not know the exact state the system is then the manager must estimate a probability distribution, known as *belief state*, over the possible states  $S$ . This estimation is used as a seed to be refined by the POMDP executions.

Figure 2(1) presents a system transiting from state  $S$  to state  $S'$ , supported by MDP. Also, Fig. 2(2) presents a diagram with a system transiting from state  $S$  to state  $S'$ , supported in a partial observation, and using a belief state to achieve the reward on  $S'$ . Without knowing the actual state  $S$  at time  $t$  (cf. Fig. 2(2)), the partial observation triggers the possibility of having one or more belief states. The challenge of solving a POMDP is to maximize the reward of a given action  $A$  achieving the state  $S'$  at time  $t + 1$ , from the belief states. In the end, a control policy will yield the greatest amount of utility over some number of decision steps. As a summary, both POMDP and MDP require a set of states, a set of actions, transitions and rewards. The actions' effects on the state in a POMDP is exactly the same as in an MDP. The difference is in whether or not we can observe the current state of the process. In a POMDP we add a set of observations to the model. So instead of directly observing the current state, we obtain an observation which provides a hint about what state it is in. The observations are probabilistic; therefore, an observation model encompassing the probability of each observation for each state in the model should be defined.

## 2.2 DEMO Theory and Methodology

From the business processes point of view, DEMO theory and methodology [5] introduces capabilities to deal rigorously with the dynamic aspects of the process-based business transactions using an essential ontology that is compatible with



**Fig. 2.** State transition from state  $S$  to state  $S'$ . MDP (1) and POMDP (2) solutions.

the communication and production, acts and facts that occur between actors in the different layers of the organization. A DEMO business transaction model [6] encompasses two distinct worlds: (i) the transition space and (ii) the state space.

On the one hand, the DEMO transition space is grounded in a theory named as  $\Psi$ -theory (PSI), where the basic transaction pattern includes two distinct **actor roles**: the Customer and the Producer. The goal of performing such a transaction pattern is to obtain a new fact. The transactional pattern is performed by a sequence of coordination and production **acts** that leads to the production of the new fact. In detail, encompasses: (i) order phase that involves the acts of request, promise, decline and quit, (ii) execution phase that includes the production act of the new fact itself and (iii) result phase that includes the acts of state, reject, stop and accept. Firstly, when a Customer desires a new product, he requests it. After the request for the production, a promise to produce the production is delivered by the Producer. Then, after the production, the Producer states that the production is available. Finally, the Customer accepts the new fact produced. DEMO basic transaction pattern aims specifying the transition space of a system that is given by the set of allowable sequences of transitions. Every state transition is exclusively dependent from the current states of all surrounding transactions. There is no memory of previous states. This memoryless property holds with Markov theories. On the other hand, the DEMO state space delivers the model for the business transactions **facts**, which are products or services, and that are obtained by the business transaction successful execution. Throughout the business transaction execution more intermediate facts are required.

Based in the stated above, we conceptualize the DEMO business transactions as a set of triples using the dimensions of: **actor roles**, **acts** and **facts**. This conceptualization could also be aligned with the Subject-oriented Business Process Management [8] work where the three core dimensions of a business processes are: **subject**, **predicate** and **object**. This possible alignment will be further assessed in detail in the near future.



methodology with (at least) the capabilities of modeling the transition, the state and actor role spaces, *e.g.*, DEMO [5], SBPM [8], BPMN [10], *etc.* Afterwards, the business transactions models are converted in a set of memory less triples as introduced previous in this section (cf. Fig. 3). The advantage of decoupling steps 1 and 2 is because is easier to find the triples after producing a business transactions model. Then, step 3,  $P$  is populated with the POMDP tuple estimation. Usually, it corresponds to a file creation in the POMDP format<sup>2</sup>. To facilitate the generation of the POMDP file (summing up to 7500 configuration lines in our case study, cf. Sect. 4) a JAVA application was specially developed for this purpose. After that, from step 4 until 11, the POMDP is computed: (1) execute the action that the current node tells us; (2) receive the resulting observation from the world; (3) transition to next node based on the observation; (4) repeat to step (1). In the end, a policy graph mapping  $Z \rightarrow A$  is delivered (cf. shadowed ellipsoid area in Fig. 2). Finally, the policy graph is rendered using any graphical tool.

---

**Algorithm 1.** Method to compute the informed decision-making.

---

**Require:** Business transaction prescriptions

**Ensure:** Control policy graph ( $Z \rightarrow A$ )

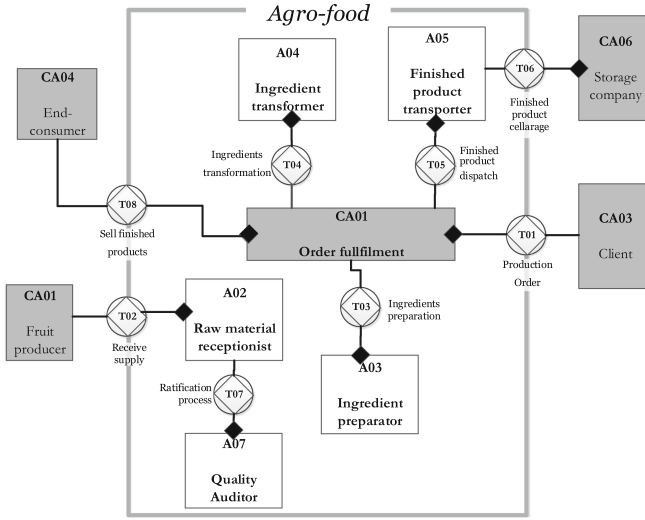
- 1: **Set**  $M \leftarrow$  Model the prescribed business transactions.
  - 2: Convert  $M$  in a set of triples:  $S_i = \langle Act_i, Fact_i, ActorRole_i \rangle$ .
  - 3:  $P \leftarrow$  POMDP tuple ( $S, A, Z, T, O, R, \gamma$ ) estimation.
  - 4: **for all** node of  $P$  **do**
  - 5:     **for each**  $Z$  **do**
  - 6:         Calculate  $Prob(Z)$
  - 7:         Calculate Belief State
  - 8:         Calculate  $\mathbf{R}$
  - 9:         Calculate  $A$
  - 10:     **end for**
  - 11: **end for**
  - 12: Render the computed policy graph ( $Z \rightarrow A$ ) using a graphical tool.
- 

## 4 Case Study

An agro-food industry company focusing the transformation of fresh fruits to preparations that are sold to other companies is considered. Its clients are industries of milk-based products, ice creams, cakes and beverages products. To guarantee the product quality, fruit producers are subject to a ratification process before starting supplying fruit. The fruit passes through three stages: (i) raw material, (ii) ingredients after raw material preparation, and (iii) finished product after ingredients transformation. Until reaching the end consumer, a complex value chain is executed including the actor roles of: client, fruit producer, raw material receptionist, ingredient preparator (*e.g.*, weighing and cleaning),

---

<sup>2</sup> An example of this standard format could be consulted at [2].



**Fig. 4.** Ator transaction diagram of Agro-food case study.

ingredient transformer (*e.g.*, mixing components, adding water, sugar or other products accordingly with the recipe), finished product transporter and storage company (when the agro-food company is not able to locally store all the production). The production starts when a client order is received (produce to order policy). Then, five stages are performed: receive supply, ingredients preparation, ingredients transformation, finished product cellaring and dispatch. Besides selling to other companies, they also sell a small part of finished products directly to the end consumer. Table 1 explains the result obtained with each business transaction. In more detail, Fig. 4 depicts the business transactions involving the actor roles by an actor transaction diagram (ATD) in DEMO [5].

Due to the value chain raising complexity, including many other companies (*e.g.*, suppliers), and also, due to the food safety legal obligations, traceability is a core functionality to identify the products throughout the production value chain. It encompasses three basic considerations: the product identification, the product origin and the product destination. When a lot infection is detected, traceability aids the identification of its location and removing it from the market. A lot infection may occur due to many workarounds, *e.g.*, recipe not followed by ingredient transformer, allergenic material infection at ingredient preparation, contamination during transportation, bad temperature conditions for transportation or fruit disruption stock.

Table 2 synthesize the POMDP variables that are estimated for this case study<sup>3</sup>. To begin with,  $S$  is given by the ATD DEMO model depicted in Fig. 4,

<sup>3</sup> The full POMDP file is public available with doi:[10.13140/2.1.4433.2326](https://doi.org/10.13140/2.1.4433.2326).



**Table 1.** Transaction product table of Agro-food case study.

Transaction kind	Product kind
T01 - Production order	P01 - Client Order CO is completed
T02 - Receive supply	P02 - Supply Order SO is completed
T03 - Ingredients preparation	P03 - Ingredients I of Client Order are prepared
T04 - Ingredients transformation	P04 - Ingredients I of Client Order are transformed
T05 - Finished product dispatch	P05 - Finished product FP of Client Order is dispatched
T06 - Finished product cellarage	P06 - Finished product FP of Client Order is stored
T07 - Ratification process	P07 - Process P is ratified
T08 - Sell finished products	P08 - Finished product FP is sold

and in detail by the DEMO business transaction space (cf. Fig. 5) where each state is grounded by the triple  $S_i = \langle Act_i, Fact_i, ActorRole_i \rangle$ . For clear explanation, the triple is simplified by the pair:  $S_i = \langle Act_i, ActorRole_i \rangle$  avoiding the fact types involved in all business transactions being. A full usage of DEMO business transaction space is explained in [12]. Nevertheless, this simplification does not affect the nature of the obtained decision-making results and consequent conclusions. In the right part of Fig. 5, four flows of work operated by the organization are identified: (i) production to client order, (ii) cellarage, (iii) fruit supply and (iv) selling products to end consumer.

Recalling Table 2,  $Z$  and  $A$  represent respectively the observation and the actions.  $Z$  differs from  $S$  in the sense that is much simpler and is totally business-oriented. Managers are able to observe if products are being delivered correctly, if any complaint was received, if stock is below a certain threshold, if any problem occurred while the products are being transported or else if it is running correctly (OK) so far. Therefore, in this example, there exist 40 possible states defined from the business transactions model (cf. Fig. 5), but only 5 possible observations may occur in operation.

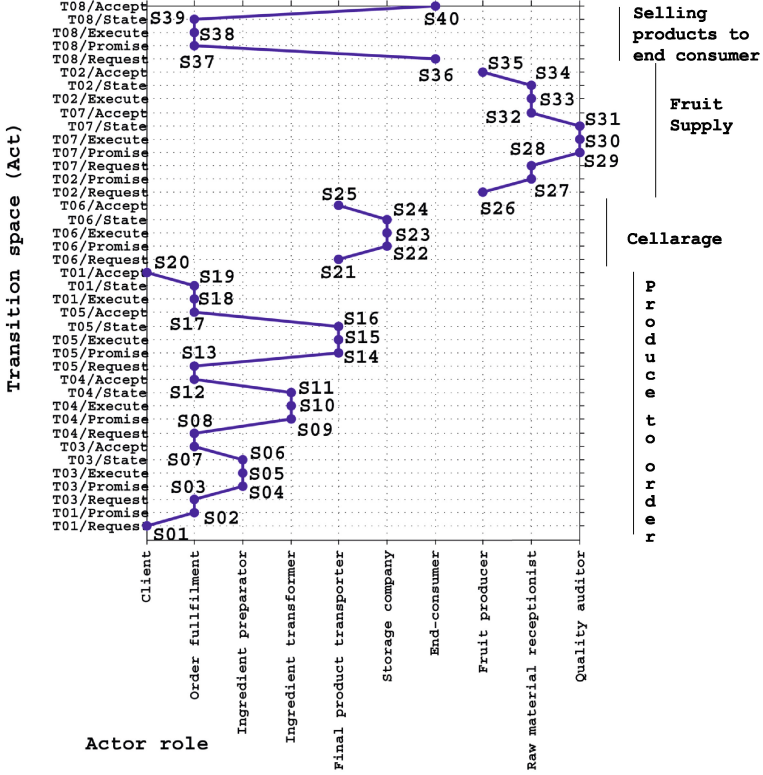
The POMDP variable  $A$  specifies the capability of management to take actions. Four distinct actions are possible: no action (*no\_op*), to cancel a previous order requested by a client (*cancel\_client\_order*), to request more ingredients from a fruit producer (*request\_more\_ingredients*) and to send quality questionnaire to the clients to assure their level of satisfaction (*send\_quality\_questionnaire*).

Regarding  $T$  matrix, for each action a probability is estimated assuming that transition from an initial to a final state occurs. On the one hand, the *no\_op* action do not have impact in the normal progress of the business transactions operation. For simulation purposes, we assumed that in 95 % of situations if *no\_op* is enacted then the states follows as described in Fig. 5. However, when action *send\_quality\_questionnaire* is enacted then a 50 % chance of sending it to the client exists if the selling business transactions are at stating transition step (S19 and S39). On the other hand, *cancel\_client\_order* and *request\_more\_ingredients* actions changes their normal operational progress. The first one, restarts the produce to order flow (jump to S01), and the second one, invokes the fruit supply flow (jump to S26). With correspondingly 90 % and 75 % chance of happening.

**Table 2.** POMDP variables definition for Agro-food case study.

POMDP variable	Value
$S$	40 states ( $S01 \dots S40$ ) following the DEMO standard pattern of a transaction: $\{ < T01\_request, Client>, \dots, < T01\_accept, Client>, \dots, < T08\_request, End\ consumer>, \dots, < T08\_accept, End\ consumer> \}$ . Full specification in Fig. 5
$Z$	$\{product\_delivered, complaint, stock\_break, transport\_disrupt, running\_ok\}$
$A$	$\{no\_op, cancel\_client\_order, request\_more\_ingredients, send\_quality\_questionnaire\}$
$T : A \times S \times S \rightarrow \prod(S)$	$A = no\_op \rightarrow$ Proceed states cf. Fig. 5 by 95 % of situations $A = send\_quality\_questionnaire$ AND $S \in [19, 39] \rightarrow$ true in 50 % of situations $A = cancel\_client\_order$ AND $S \in [1..20] \rightarrow$ restart $S01$ <b>else</b> proceed states cf. Fig. 5 by 90 % of situations $A = request\_more\_ingredients$ AND $S \in [1..20] \rightarrow$ invoke fruit supply ( $S26$ ) <b>else</b> proceed states cf. Fig. 5 by 75 % of situations
$O : A \times S \times Z \rightarrow \prod(Z)$	Regarding the end state of producing and selling products: $S = (20 \text{ OR } 40)$ AND $Z = product\_delivered \rightarrow 70 \%$ $S = (20 \text{ OR } 40)$ AND $Z = stock\_break \rightarrow 10 \%$ $S = (20 \text{ OR } 40)$ AND $Z = complaint \rightarrow 10 \%$ $S = (20 \text{ OR } 40)$ AND $Z = (transport\_disrupt \text{ OR } running\_ok) \rightarrow 5 \%$ Regarding cellarage end state: $S = 23$ AND $Z = transport\_disrupt \rightarrow 20 \%$ <b>else</b> 80 % Regarding all other end states: $Z = running\_ok \rightarrow 95 \%$ <b>else</b> 5 %
$R : A \times S \times S \times Z \rightarrow \mathbf{R}$	Flow of work ends: $\{S20, S25, S35, S40\} \rightarrow \mathbf{R} = 1$ $complaint \text{ OR } stock\_break \text{ OR } transport\_disrupt$ are observed $\rightarrow \mathbf{R} = -5$
$\gamma$	5 %
Start state	$S01$
Start action	$request\_more\_ingredients$

Regarding the  $O$  matrix, for all  $A$  that moves to end state  $S$  it delivers an observation  $Z$  with probability  $P$ . The estimation follows the reasoning: in the majority of the situations (95 %) *running\_ok* is observed. When the end state of producing ( $S20$ ) or selling the products ( $S40$ ) is achieved, the observations of *stock\_break* or *complaint* could happen with 10 % probability each. Also, when the cellarage is being executed ( $S23$ ) the *transport\_disrupt* could be observed with 20 % probability, *e.g.*, when a truck has an accident.



**Fig. 5.** Model of business transaction space for Agro-food case study (40 states). Left axis: acts, right axis: flows of work, and bottom axis: actor roles.

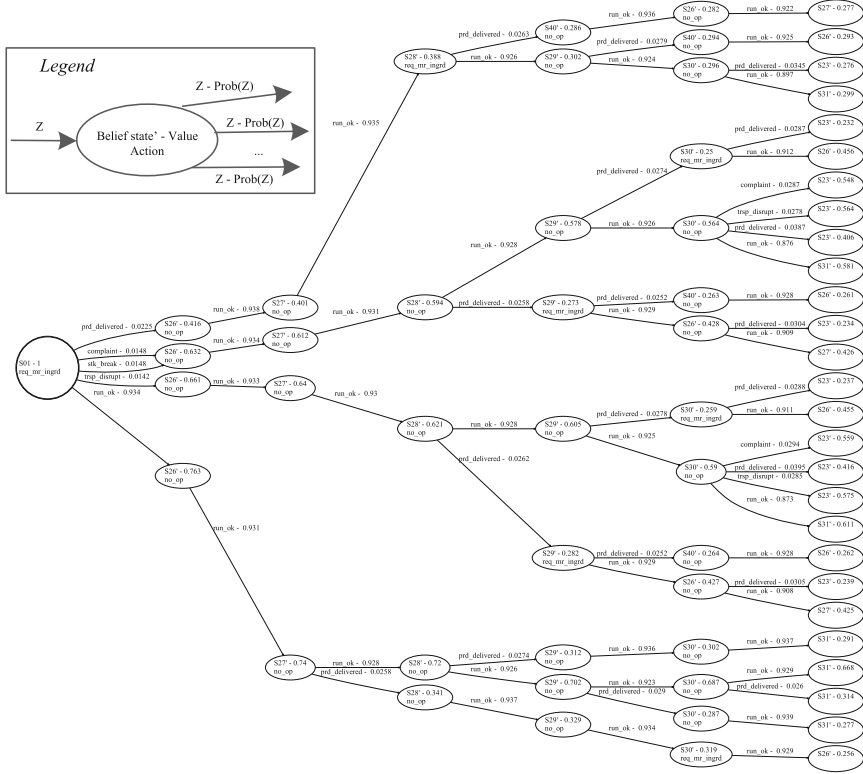
Finally, regarding  $R$  matrix, if any flow of work is terminated successfully then a reward of 1 is assigned. Otherwise, a negative reward of  $-5$  is used as a penalty.

## 5 Qualitative Evaluation

We operationalized our proposal for the Agro-food case study by applying MDP and POMDP solvers. The POMDP was performed cf. Algorithm 1 where the solver is APPL toolkit [2], which is a recent C++ implementation running in Linux environment<sup>4</sup>. The delivered policy graph is rendered using GraphViz [9] tool. The MDP is computed by a *Matlab*<sup>®</sup> toolbox<sup>5</sup> using a linear programming algorithm. The intent of our proposal is to explore the benefits of using stochastic approaches to aid the management decisions. This goal can be achieved if

<sup>4</sup> Others POMDP solvers are available, *e.g.*, Perseus [22] implementation of randomized point-based approximate value, Tony Cassandra [4] solver, the ZMDP solver for POMDP and MDP [21].

<sup>5</sup> Toolbox public available at <http://www7.inra.fr/mia/T/MDPtoolbox>.

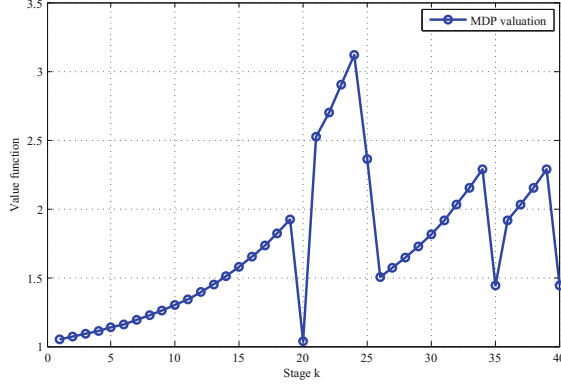


**Fig. 6.** Agro-food POMDP control policy,  $\text{graph-max-depth}=6$ ,  $\text{graph-min-prob}=2.5\%$ .

engineers are empowered with full pertinent information to forecast the impacts of their decisions in the near future of the organization.

On the one hand, a POMDP delivers a policy graph mapping the observations ( $Z$ ) into actions ( $A$ ), maximizing the reward, and yielding the greatest amount of utility over the different decisions through out a time-wide horizon. Whenever a decision is needed, this policy graph guides the engineer. By other words, it serves as a decision map. Unlike the usual challenge of finding the best path in a graph, our solution offers the graph to be followed by the organization. Figure 6 depicts the policy graph with time horizon concerning 6 consecutive observations, whereas the occurring probability is greater than 2, 5 % (except for the first run).

An ellipse represents a belief state ( $S'$ ) that is reached by taking an action ( $A$ ) and a branch represents an observation ( $Z$ ). Given the initial state and action the graph follows from the left to the right side, expanding the different actions that are recommended as a reaction for each observation. The value represented in each branch is the probability of occurring a given observation ( $\text{Prob}(Z)$ ). The value represented in each ellipse is the actual reward value ( $R$ ) of taking that path throughout execution. Regarding the results delivered by Fig. 6, we identify that the actions that maximize the utility of this specific configuration is given by the path through the actions *no\_op*. After 6 consecutive *run\_ok* observations



**Fig. 7.** Agro-food MDP valuation. Each stage is an observable state ( $S01...S40$ ).

the belief state is  $S31$  and the  $\mathbf{R} = 0,668$ . However, in the case of occurring other observations, Fig. 6 fully describe the rewards for the future actions that should be taken in order to obtain a local maximization.

On the other hand, a MDP solves the problem of calculating an optimal policy in an accessible and stochastic environment with a known transition model. Figure 7 delivers the result of valuating the execution of all consecutive states ( $S01..S40$ ) when each state correspond directly to an observation.  $S$ ,  $A$ ,  $R$  and  $\gamma$  hold with the definitions contained in Table 2. The transition state  $T$  is simpler because only one action is recognized for each state transition. For simulation purposes, the *no\_op* action has been considered. In this experimental setup, we find that if no action is taken and the business transactions follows as prescribed the value rise along with the execution of the flows of work.

Comparing POMDP and MDP solutions in terms of benefits for the engineering of decision-making, we find that two different purposes are fulfilled. First, the POMDP results are mapped in a time-wide horizon that forecast the probabilistic belief states from the observations and enacted actions. This result allows the business manager to focus in a black-box perspective of the organization, supporting the decision-making process with more information, even when the business processes are not fully observable. In addition, the business managers are able to dynamically regenerate their strategic plans, whenever any estimation variable or organizational dynamics change. Second, a MDP forecasts local (and global) valuation for business transactions execution assuming that business processes are fully observable. Applying MDP to different business processes design decisions, the optimal (and sub-optimal) solutions that meet the organizational goals could be anticipated prior to its implementation.

## 6 Conclusions and Future Work

The aim of this paper is to contribute to solve the problem of organizational decision-making (*e.g.*, business processes re-engineering) in partially known environments (usually named by partially observable). Specifically, this paper

addresses the environments of business processes execution that are supported by enterprise information systems, which by its turn, are complex and partially observable. The solution support the management decisions, providing maps that express the impacts of management decisions on the organizational operation. Therefore, it minimizes the risk of making wrong decisions (*e.g.*, incorrect change of business processes) and power up a positive impact on the national economy services industry.

To obtain this result, we analyzed the contemporary problems for decision-making and designed a novel solution that combines DEMO-based business process design and operation research (MDP and POMDP Markov theories). In the daily operation, manager and engineers take decisions that are based upon the available observations at each instant in time. Because of partially available information, these observations do not fully describe the actual state of the organization and impose to the manager the problem of guessing what state it is in.

Our solution valuates the actions that could be enacted from the available partial observations, using a probabilistic approach, where an initial estimation effort for the tuple  $(S, A, Z, T, O, R, \gamma)$  is demanded. In the end, managers and engineers are empowered with full pertinent information to forecast the impacts of their decisions in the near future of the organization.

Future work will involve two main threads of work: *(i)* the technical integration between the Markov theories and DEMO theory and methodology. On the one hand, the results delivered by this work will benefit from the theoretical POMDP and MDP advances regarding the algorithmic performance optimization and from all the aspects related with fast-approach solution convergence. On the other hand, the estimation of  $T$  and  $O$  matrices is actually a complex task that demands the development of automatic tools; *(ii)* more case studies are needed in order to achieve a broader generalization of the results and more empirical findings.

**Acknowledgments.** This work was supported by Project nr 652643 (Respon-SEAble), under the title: “*Sustainable oceans: our collective responsibility, our common interest. Building on real-life knowledge systems for developing interactive and mutual learning media*” from H2020 programme.

## References

1. Alter, S.: Theory of workarounds, communications of the association for information systems. Univ. Access Inf. Soc. **34**(55), 1041–1066 (2014)
2. Cai, P.: Approximate pomdp planning (appl) toolkit (2010). <https://github.com/petercaiyyo/appl>
3. Cassandra, A.R.: A survey of POMDP applications. In: Working Notes of AAAI 1998 Fall Symposium on Planning with Partially Observable Markov Decision Processes, pp. 17–24 (1998)
4. Cassandra, A.R.: Pomdp solver, v53 (2005). <http://www.pomdp.org/code/index.shtml>
5. Dietz, J.L.G.: Enterprise Ontology: Theory and Methodology. Springer, Heidelberg (2006)

6. Dietz, J.L.: The deep structure of business processes. *Commun. ACM* **49**(5), 58–64 (2006). <http://doi.acm.org/10.1145/1125944.1125976>
7. Dietz, J.L., Hoogervorst, J.A., Albani, A., Aveiro, D., Babkin, E., Barjis, J., Caetano, A., Huysmans, P., Iijima, J., van Kervel, S., et al.: The discipline of enterprise engineering. *Int. J. Organ. Des. Eng.* **3**(1), 86–114 (2013)
8. Fleischmann, A., Stary, C.: Whom to talk to? a stakeholder perspective on business process development. *Univ. Access Inf. Soc.* **11**(2), 125–150 (2012). doi:[10.1007/s10209-011-0236-x](https://doi.org/10.1007/s10209-011-0236-x)
9. Graphviz: Graph visualization software (2014). <http://www.graphviz.org/>
10. Group, O.M.: Bpmn specification 1.2. pdf available in internet, January 2009
11. Guerreiro, S.: Decision-making in partially observable environments. In: 2014 IEEE 16th Conference on Business Informatics (CBI), vol. 1, pp. 159–166, July 2014
12. Guerreiro, S.: Business rules elicitation combining markov decision process with demo business transaction space. In: 2013 IEEE 15th Conference on Business Informatics (CBI), pp. 13–20 (2013)
13. Guerreiro, S., Tribolet, J.: Conceptualizing enterprise dynamic systems control for run-time business transactions. In: ECIS, p. paper 5 (2013)
14. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. *MIS Q.* **28**(1), 75–105 (2004). <http://dl.acm.org/citation.cfm?id=2017212.2017217>
15. Hoogervorst, J.A.: Enterprise Governance and Enterprise Engineering. The Enterprise Engineering Series, Springer Science (2009)
16. Laudon, K., Laudon, J.: MIS - Management Information Systems: Managing the digital firm. Pearson, Upper Saddle River (2013)
17. Montiel, L., Bickel, J.: A simulation-based approach to decision making with partial information. *Decis. Anal.* **9**(4), 329–347 (2012)
18. Puterman, M.L.: Markov Decision Processes: Discrete Stochastic Dynamic Programming. Wiley, New York (1994)
19. Russell, S., Norvig, P.: Artificial Intelligence: A Modern approach, 3rd edn. Pearson, London (2009). ISBN-13: 978-0136042594
20. Shewhart, W.: Economic Control of Quality of Manufactured Product/50th Anniversary Commemorative Issue. American Society for Quality, June 1980
21. Smith, T.: Zmdp software for pomdp and mdp planning (2007). <https://github.com/trey0/zmdp>
22. Spaan, M.: Perseus randomized point-based approximate value iteration algorithm (2004). <http://users.isr.ist.utl.pt/mtjspaان/software/index.en.html>
23. Vanderfeesten, I., Reijers, H.A., van der Aalst, W.M.P.: Product-based workflow support. *Inf. Syst.* **36**(2), 517–535 (2011). doi:[10.1016/j.is.2010.09.008](https://doi.org/10.1016/j.is.2010.09.008)
24. Weber, M.: Decision making with incomplete information. *Eur. J. Oper. Res.* **28**, 44–57 (1987)

Advances in Enterprise Engineering IX  
5th Enterprise Engineering Working Conference, EEWC  
2015, Prague, Czech Republic, June 15-19, 2015,  
Proceedings  
Aveiro, D.; Pergl, R.; Valenta, M. (Eds.)  
2015, XV, 161 p. 47 illus., Softcover  
ISBN: 978-3-319-19296-3