Consistent Integration of Decision (DMN) and Process (BPMN) Models

Laurent Janssens[1,2], Ekaterina Bazhenova[3], Johannes De Smedt[1], Jan Vanthienen[1], and Marc Denecker[2]

[1] KU Leuven, Leuven Institute for Research on Information Systems, Leuven, Belgium
{laurent.janssens,johannes.desmedt,jan.vanthienen}@kuleuven.be
[2] KU Leuven, Declarative Languages and Systems, Leuven, Belgium
{laurent.janssens,marc.denecker}@kuleuven.be
[3] University of Potsdam, Hasso Plattner Institute, Potsdam, Germany
ekaterina.bazhenova@hpi.de

Abstract. Organizations use business process and decision management techniques to run their businesses more efficiently. Modeling the business processes and decisions is a vital part of this. The recently proposed Decision Modeling and Notation (DMN) standard introduces a declarative approach for modeling decisions and aims at the separation of decision logic from business processes. Decoupling decisions and process models is crucial for the flexibility and maintainability of business processes. However, this aspect has received little attention, except in straightforward situations. In this paper we identify five integration scenarios and provide a formal basis for DMN models. Using these scenarios and formalization we illustrate how to achieve consistent integration of decision and process models.

Keywords: Decision Modeling, Process Modeling, BPMN, DMN

1 Introduction

Business process management (BPM) and decision management (DM) are being used to improve the efficiency and effectiveness of organizations. Companies are interested in running effective and competitive processes, and use BPM to describe and improve these processes. The BPMN standard [12] allows processes to be described visually in a structured, and executable way. However, while decision making is an important aspect of BPM, BPMN supports no clear-cut way to represent this. Similarly decision management is being used to map the decisions made within the business, and to evaluate, and improve these decisions. However, there is little emphasis on how these decisions are made in a process context. Moreover, complex decisions are often modeled as processes to guide the business through the decision making process. This can result in difficulties to maintain and redesign decisions, as mentioned in [4].

Recently the emphasis has been on integrating decision and process management. The DMN standard [13] was developed to support decision modeling complementary to processes represented using BPMN. In this paper we describe five integration scenarios, which allow us to identify situations where integration plays an important role. For two of these scenarios we discuss the challenge of achieving consistent integration, i.e. integration between a decision and process model which preserves a consistency with respect to information requirements. To support our approach we introduce a formalization of decision requirements in DMN.

2   Background

Recently the separation of processes and decision logic has become an evident trend. In response to this trend the OMG group developed the new Decision Model and Notation (DMN) [13] standard as a standard approach for modeling decisions within organizations [15]. At the same time emphasis is placed on integrating decision management and process management.

The DMN standard allows to model and describe decisions in a declarative way on two levels, the requirements level and decision logic level. For the first level decisions requirement diagrams (DRD) are used to represent the information requirements of the decisions in the model. These diagrams can consist of several types of elements, decisions, input data, business knowledge models, and knowledge sources. In this paper we take abstraction of knowledge models and sources, since our approach relies solely on the decisions and input data. Information requirements in the DRDs represent the requirements of decisions in terms of subdecisions and input data, depicted using arrows going from the requirement to the decision. An example of a DRD is given in Figure 1.
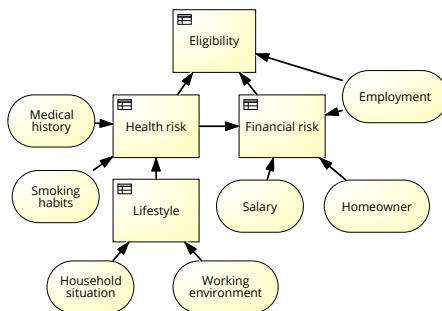

Fig. 1. Decision requirement diagram for the credit eligibility decision

The second level uses the declarative FEEL expression language to describe the decision logic behind every decision. DMN provides no guarantees for efficient resolution of decisions, this is left to the invoking context. Additionally

decisions have no side-effects and storage of outputs and intermediate results is unconstrained and vendor-dependent. DMN is designed to be complementary to the BPMN standard, as discussed in [5].

In the trend towards integration several situations can be identified. Basic solutions see processes represented using only BPMN, or decisions using only DMN. In other cases decisions are often emulated using intricate process control flows, which can result in cascading gateways. These hidden decisions must be identified in the process. After identifying and modeling these decisions the resulting model must be integrated consistently with the process model.

In more complex processes several decisions might influence the flow and result. Correct representation and invokation is crucial for a proper understanding. Since the decisions' context can have an impact on their results information requirements have to be taken into account when designing the process. This contrasts the declarative nature of DMN. Here true integration of process and decision models is required. However, these situations have received little attention. By integrating decision models and process models, new challenges arise. One of these challenges is ensuring the consistency between the decision models and associated process models.

3   Consistent Integration of DMN and BPMN Models

3.1   Integration Scenarios

When considering decisions in processes five scenarios can be identified. This section outlines each of these scenarios, discussing their occurrence and the possible relation between them. Challenges for each scenario are identified and existing solutions discussed where appropriate.

Scenario 0. Processes without Decisions In cases where a process consists of a series of predetermined activities and the control flow is simple. Thus, there is no opportunity for integration with a decision model, since there are no decisions to be made. Often this will be subprocesses of more complex business processes.

It is important to note, processes where decisions are hard-coded using complex control flows, are bad examples of this scenario. Typically, changing these decisions is difficult as process paths have been fixed, making it hard to re-evaluate the outcomes and factors for invoking the decision [16].

Scenario 1. Local Decisions In this scenario local decisions ensure simple separation of control flow and decision logic. These decisions, characterized by their atomic outcome and local effect, can still be complex and require multiple subdecisions. However they have no impact on the rest of the process and can hence be inserted safely. Decision management is handled by providing decision outcomes as inputs for the process control flow. This simplifies and increases the flexibility of the process. These models can be obtained from mining techniques [1, 9].

Scenarios 2 & 3. Processes with Interrelated Decisions The second and third scenarios include situations where multiple interrelated decisions are made during the process. Often both the flow and result of the process will depend heavily on the outcomes of these decisions.

In the second scenario the decisions are placed directly before the business activities requiring their output, thus Here the information pertaining to the relation of the decisions is kept externally in the decision model. This constitutes an inefficient ordering of the decisions and requires additional data management, to store results of subdecisions.

In the third scenario the structure of the decisions is used while modeling the process. By ordering related decisions according to their requirements no decision has to be made twice, increasing the efficiency of the process.

For process models fitting either of these scenarios it is crucial to have a decision model, since information on the relation between the different decisions is required for understanding and executing the process. In both of these scenarios it is important that the process model is consistent with the decision model, discussed further in Section 3.4.

Scenario 4. Knowledge Intensive Processes The last scenario constitutes the opposite situation as Scenario 0, here the process is actually a single decision. This scenario occurs when a process is highly knowledge intensive, i.e. when the process is used to decide a single output. In these cases only a decision model is required. As in Scenario 0 integration offers little benefit in this situation.

3.2   Example Case

In our example, customers can apply for credit at the company, thus starting a new process instance. After receiving the necessary information a decision has to be made about whether the customer is eligible for a loan. This decision is based on the customer's health and financial situation, to minimize the risk of non-payment. The requirements for this decision are illustrated using DMN in Figure 1. If the customer is eligible for a loan an insurance is filed and the loan accepted, if not the loan is rejected.

The credit application process is depicted using BPMN in Figure 2. The process contains two business activities, the eligibility decision, and filing the insurance. However, the decision requirement diagram in Figure 1 shows the two data objects `Health risk category` and `Financial risk category` are actually the respective outputs of the `Health risk` and `Financial risk` decisions. Using these intermediate results in the process model poses problems for flexibility and maintainability. If the structure of the `Eligibility` decision changes and these decisions are no longer needed, the process model will necessarily need to change. To resolve this issue, the decisions determining these intermediate results should be added to the process model.
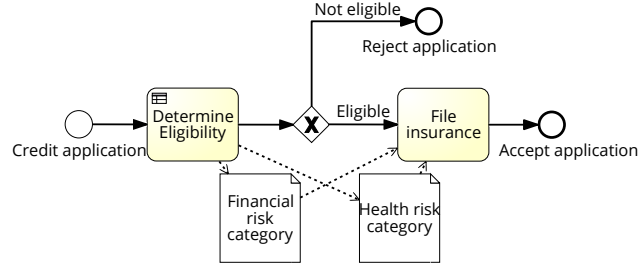
Fig. 2. Business process where intermediate decision results are used

3.3   Formalization

To support our approach we introduce a formal basis for decisions and requirements in DMN models.We take abstraction from the use of Business Knowledge Models and Knowledge Sources. However, all definitions and theorems provided can be readily extended to include the use of these concepts.

We adopt the definition of a *decision requirement diagram* from [2] and slightly extend it to identify an important constraint, i.e. that no cycles can occur in the information requirements.

Definition 1. *A decision requirement diagram DRD is a tuple* (D_{DM}, ID, IR) *consisting of a finite non-empty sets of decision nodes* D_{DM}*, input data nodes* ID*, and directed edges* IR *representing the information requirements such that* $IR \subseteq D_{DM} \cup ID \times D_{DM}$*, and* $(D_{DM} \cup ID, IR)$ *is a directed acyclic graph.*

In the service-oriented approach a decision is implemented as a service offering a single decoupled point of entry to the business logic of that decision. This has benefits for automation and flexibility, since the process only needs information about the service's interface. [11]. To define the interface of a decision we define the input requirement set and a decision's output set in Definition 2.

Definition 2. *The interface of a decision D is a tuple* $(dirs_D, O_D)$*. Where the decision input requirement set* $dirs_D$ *is the set of input data directly or indirectly required by D. The output set* O_D *is the set of all possible outputs of D.*

While the input is clearly indicated in the DRD, as in Figure 1, outputs are not specified explicitly. They are made explicit on the decision logic level. In this case we assume the output is identified by the decision's name.

To identify incorrect uses of decisions in process models it is important to know their structure and meaning. As in the example in Figure 1 decisions are often structured to use the results of other intermediate decisions, called subdecisions. The subdecision outputs are defined as intermediate results in Definition 3.

Definition 3. *An output O is an intermediate result of decision D if and only if* $O \notin O_D$ *and there exists a subdecision* D' *of D for which* $O \in O_{D'}$*.*

In the context of DMN, the execution of a decision is called invoking that decision. Definition 4 defines when a decision can be invoked, based on the available data.

Definition 4. *A decision D is invokable from a set of data elements S if it contains all the required inputs, $dirs_D \subseteq S$.*

This formal framework will allow us to identify inconsistencies between a decision model and an associated process model. The process of identifying and resolving these inconsistencies is described in the following subsection.

3.4   Consistent Integration for Scenarios 2 and 3

Using the formalization from the previous subsection, we can define when a process model fitting integration scenario 2 or 3 is consistent with an associated decision model. Definition 5 defines consistency based on the availability of required input and the use of decision outputs.

Definition 5. *A business process model is consistent with a decision model iff no intermediate results of invoked decisions are used, and each decision invoked in the process, must be guaranteed to be invokable at that process stage.*

When addressing inconsistencies we assume changing a decision model will, in most cases, change the decisions' results, while changing the order of activities of a process in a systematic way will have little or no impact on the result. Under this assumption our approach uses the decision model as a reference for consistent integration.

Resolving the Use of Intermediate Results The process model in Figure 2 violates the first condition of Definition 5. Intermediate results of the `Eligibility` decision are used, i.e. the process model is inconsistent with the decision model in 1. We identify the `Health risk category` and `Financial risk category` as being intermediate results, produced by the `Health risk` and `Financial risk` decisions. Thus, these decisions must be added to the process model. The correct decision order differs based on the scenario. Conformance with scenario 2 requires the two decision activities producing the intermediate results to be placed directly in front of the `File insurance` activity. Conformance with Scenario 3, is achieved when the requirement structure of the decision model is used to determine the order of the decisions. The main benefit of invoking these decisions explicitly in the model is flexibility. Should the `Eligibility` decision change, so that the subdecisions are no longer relevant, this will not affect the process. The resulting process, conformant with scenario 3, is shown in Figure 3.
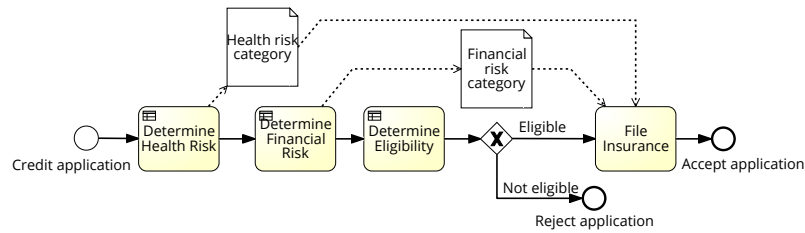
Fig. 3. Consistent process model, conformant with scenario 3

4   Related Work

Ensuring the consistency of process models using data has been investigated in numerous works already. Many data-aware process modeling approaches have been proposed that use different types of data representation and input, such as the ontology-based knowledge-intensive approach in [14], enhancing DMN and declarative process models [10] and all works concerning colored Petri nets [8]. The latter offers a complete formalized approach to deal with integration, sticking mainly to local data. Approaches such as [3] use Petri nets as they offer ways to ensure consistency between data and process model. However, the focus on data often downplays the holistic view that should be achieved to support decisions, and does not deal with DMN.

The separation of concerns has enjoyed plenty of attention, mainly in the domain of software modeling and design [7]. They offer firm motivation for keeping multi-perspective modeling tasks, such as control flows and decision making, isolated. The externalization of business logic was already discussed in [6] in the context of business rules. This need for separation carries over to decision logic. In the field of process mining this attention to separation is equally important. In [1] a method is described to extract the underlying decisions and simplify the corresponding process models.

5   Conclusion and Future Work

In this paper we identify five modeling scenarios for the combination of decision and process models. Two scenarios are identified where consistent integration of decision and process models is vital to support the separation of concerns, and related benefits to flexibility and maintainability. These scenarios include situations where interrelated decisions are invoked within the process. For these two scenarios we outline an approach which allows to identify and resolve inconsistencies between the decision and process model. To support our approach we introduce a formal basis for decisions and information requirements in DMN models. Using this formalization our approach offers a systematic way to integrate process and decision models, even for complex combinations of decisions and process activities.

In our approach only the models are used, no translations or additional data are required. This would allow it to be be introduced in modeling tools and used as a guide while modeling, to guard the user from introducing inconsistencies between process and associated decision models. As future work, we plan to extend the introduced formalization of DMN and use this to further examine the integration of business process and decision models, with added emphasis on the data perspective.

References

1. Batoulis, K., Meyer, A., Bazhenova, E., Decker, G., Weske, M.: Extracting decision logic from process models. In: Advanced Information Systems Engineering, pp. 349–366 (2015)
2. Bazhenova, E., Weske, M.: Deriving decision models from process models through enhanced decision mining (2015)
3. Cabanillas, C., Resinas, M., Ruiz-Cortés, A., Awad, A.: Automatic generation of a data-centered view of business processes. In: Advanced Information Systems Engineering, pp. 352–366 (2011)
4. De Roover, W., Vanthienen, J.: On the relation between decision structures, tables and processes. In: On the Move to Meaningful Internet Systems: OTM 2011 Workshops. pp. 591–598. Springer (2011)
5. Debevoise, T., Taylor, J.: The MicroGuide to Process Modeling and Decision in BPMN/DMN. CreateSpace Independent Publishing Platform (2014)
6. Goedertier, S., Vanthienen, J.: Compliant and flexible business processes with business rules. In: 7th Workshop on Business Process Modeling, Development and Support at CAiSE'06. pp. 94–104. Presses Universitaires de Namur (2006)
7. Gordijn, J., Akkermans, H., Van Vliet, H.: Business modelling is not process modelling. In: Conceptual modeling for e-business and the web, pp. 40–51. Springer (2000)
8. Jensen, K.: Coloured petri nets. Springer (1987)
9. de Leoni, M., van der Aalst, W.M.: Data-aware process mining: discovering decisions in processes using alignments. In: Proceedings of the 28th annual ACM symposium on applied computing. pp. 1454–1461. ACM (2013)
10. Mertens, S., Gailly, F., Poels, G.: Enhancing declarative process models with dmn decision logic. In: Enterprise, Business-Process and Information Systems Modeling, pp. 151–165. Springer (2015)
11. Mircea, M., Ghilic-Micu, B., Stoica, M.: An agile architecture framework that leverages the strengths of business intelligence, decision management and service orientation. Business Intelligence-Solution for Business Development (2011)
12. OMG: BPMN (2011), http://www.omg.org/spec/BPMN/2.0/
13. OMG: DMN (2015), http://www.omg.org/spec/DMN/1.0/
14. Rao, L., Mansingh, G., Osei-Bryson, K.M.: Building ontology based knowledge maps to assist business process re-engineering. Decision Support Systems 52(3), 577–589 (2012)
15. Taylor, J., Fish, A., Vanthienen, J., Vincent, P.: Emerging standards in decision modeling. BPM and Workflow Handbook series (2013)
16. Vanthienen, J., Caron, F., De Smedt, J.: Business rules, decisions and processes: five reflections upon living apart together. In: Proceedings SIGBPS Workshop on Business Processes and Services (BPS'13). pp. 76–81 (2013)