Integrating Processes, Cases, and Decisions for Knowledge-Intensive Process Modelling

Faruk Hasić*, Linus Vanwijck, and Jan Vanthienen

Leuven Institute for Research on Information Systems (LIRIS), KU Leuven
faruk.hasic;jan.vanthienen@kuleuven.be

Abstract. Knowledge-intensive processes require flexibility and scalability in modelling, as well as profound integration of data and decisions into the process. Business Process Model and Notation (BPMN) is a pertinent modelling method for processes. Until lately decisions were regularly modelled as a part of the process model in intertwined paths and gateways, negatively affecting the maintainability, comprehensibility and flexibility of processes as well as decisions. The recent introduction of the Decision Model and Notation (DMN) standard provides an opportunity for shifting in favour of a Separation of Concerns between the decision and process model. Likewise, the Case Management Model and Notation (CMMN) standard provides a method for modelling loosely structured processes in the form of cases. These three Object Management Group (OMG) standards are developed to be mutually intelligible. Previous work discusses for which modelling endeavours the separate methods should be employed. However, a clear approach towards consistently integrating the process, case, and decision concerns has still not been proposed. In this paper, we shed a light on the importance of the separation of concerns and identify inconsistencies that might arise in terms of integrated process, case, and decision modelling. Additionally, we provide a first attempt at guidelines aiming to remedy potential incompatibilities and inconsistencies.

Keywords. Process Modelling, BPMN, Decision Modelling, DMN, Case Modelling, CMMN, Integrated Modelling, Separation of Concerns

1 Introduction

An increased interest in modelling cases and decision is present in scientific work in the field of process management, as illustrated by the vast body of recent literature on Decision Model and Notation (DMN) [1–5], and Case Management Model and Notatin (CMMN) [6–12]. With these two recently introduced OMG standards it has become possible to externalise decisions and cases from the master process and to model them separately according to the *Separation of Concerns* paradigm, hence enhancing the understandability, scalability, and maintainability of processes, as well as the underlying decisions and case models.

* Corresponding author.

Decision Model and Notation is a standard for modelling decisions. DMN consists of two levels. Firstly, the decision requirement level in the form of a Decision Requirement Diagram (DRD) is used to portray the requirements of decisions and the dependencies between the different constructs in the decision model. Secondly, the decision logic level is used to specify the underlying decision logic. The standard also provides an expression language S-FEEL (Simple Friendly Enough Expression Language), as well as boxed expressions and decision tables for the notation of the decision logic. Representing decision logic in decision tables is a core concept in DMN. Decision tables have extensively been adopted in previous works, as shown in [13]. The DRD depicts decisions and sub-decisions, business knowledge models, input data, and knowledge sources. The decision logic is usually represented in the form of decision tables. A link can be made between a decision task in the process model and the actual decision model. A simple example of a DRD is given in Figure 1, representing the credit eligibility decision. DMN is a declarative decision language. Hence, DMN provides no decision resolution mechanism, this is left to the invoking context. The same holds for the processing and storage of outputs and intermediate results. This is a burden of the invoking entity (e.g. the process).

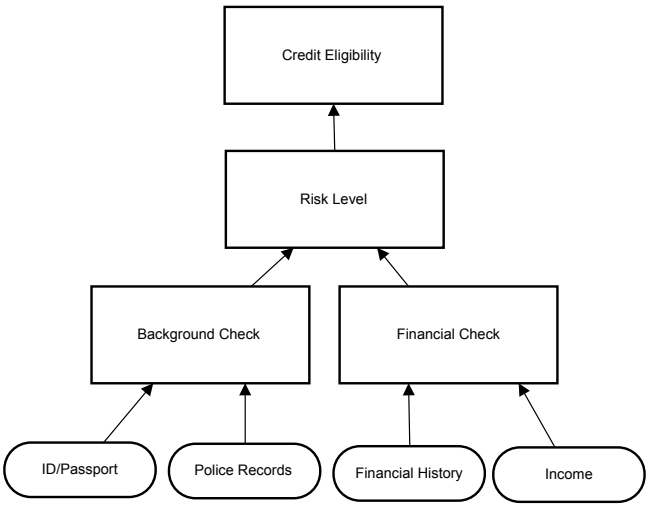

Fig. 1: A decision model for credit eligibility.

CMMN is a standard for modelling loosely structured processes or parts of processes, i.e. cases. Cases are usually resolved ad hoc, requiring flexibility during run-time in terms of planning, executing and ordering of tasks. Hence, cases are more about unstructured, non-routine work done by knowledge workers, or activities that are not predefined, depending on the situation at hand. This is in contrast with a structured and routine process such as those modelled with

BPMN. Instead of a procedural character, cases are more suitable for declarative and loosely structured flows. Typical applications for case modelling revolve around knowledge-intensive and non-predictive areas such as patient care in health and hospital processes. Figure 2 gives an example of a CMMN model for claims handling. The model exists of one *case file*, i.e *Claims File*, which contains *stages*, *events*, *milestones*, *(discretionary) tasks*, *entry* and *exit criteria*, and *planning tables*. For a full understanding of these constructs, we refer to the Case Management Model and Notation standard specification [6].

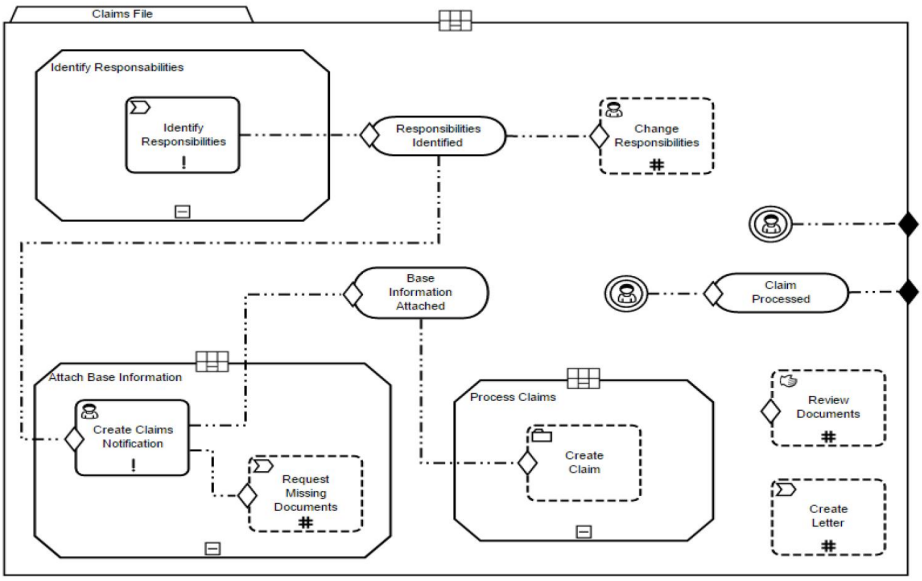

Fig. 2: A case model for claims handling [6].

This paper is structured as follows. In Section 2, a brief overview of relevant decision and case modelling approaches from literature is provided, followed by Section 3 where the separation and integration of modelling concerns is elucidated. Section 4 outlines challenges of integration by providing possible inconsistency concerns which should be remedied in future work. This section discusses the integration from a pairwise point of view, i.e. processes and decisions, processes and cases, and finally cases and decisions. Inconsistencies are enumerated and remedies for the inconsistencies are suggested. Finally, Section 5 provides conclusions and possibilities for future work.

2 Relation with Previous Work

DMN was meant mainly for business users and both the scientific and business communities have given quite some attention to the DMN standard. However, most research in the field revolves around automated discovery of decision models [4,14]. Before DMN, works revolving around process-data consistency have already been proposed. Extensions regarding data-awareness in process modelling have been researched as well. In [15] an ontology-based knowledge-intensive approach is suggested, while [16] proposes an enhancement of declarative process models [17] with DMN logic. Furthermore, countless works concerning data-aware coloured Petri Nets are available as well, offering a formally sound approach to data and process integration, such as in [18]. However, merely focusing on data fragments is not sufficient to holistically incorporate decision-awareness in processes, which DMN aims to achieve.

The decision modelling approaches present in process management literature often breach the separation of concerns between control and data flow, resulting in cascading gateways and spaghetti-like processes, hence negatively influencing maintenance and reusability [14, 19]. They do this by hard coding and fixing the decisions in processes [20]. Consequently, splits and joins in processes are misused to represent typical decision artifacts such as decision tables. Recently, more attention was given to the separation of processes and decision logic, as such an approach is supported by the DMN standard that can be used in conjunction with BPMN [21] and CMMN. Decoupling decisions and processes to stimulate flexibility, maintenance, and reusability, yet integrating decision and process models is therefore of paramount importance [3, 22–24].

When it comes to case modelling, little research has been performed so far on the topic [6–12], especially when it comes to defining how case models are related to processes and decisions. Other declarative process notations and their relation to data and decisions were discussed in literature, such as the Declare language and its extensions for data- and decision-awareness [16, 25–27].

3 Separation and Integration of Concerns

Literature has focused little on soundly integrating process and decision models. The DMN standard was developed with the apparent aim to be used in conjunction with BPMN [2, 3, 20]. Since the establishment of DMN as a standard, the general consensus is to model decisions and decision logic outside business processes. The Business Process Management field is shifting in favour of this *separation of concerns* paradigm by exteriorising the decisions and the decision logic from the process flow. Numerous tool developers have constructed new or adapted their existing tools to support the DMN standard: IBM, Signavio, Camunda, Decision Management Solutions, and FICO among others. Academic tools allowing for the modelling of BPMN, CMMN, and DMN are available as well [11, 12].

Utilising case models with processes and decisions has been addressed in literature as well [11, 12], illustrating that business, cases, and decisions are three

separate concerns that can influence each other, and hence need a clear framework on how to cooperate and consistently communicate with one another. Business Process Model and Notation (BPMN) is concerned about processing, how things should be done, i.e. the actual procedura work flow, while Case Management Model and Notation (CMMN) is about managing a particular context in terms of cases where activity execution depends on run-time circumstances, thus providing a certain degree of flexibility at execution time. Decision Model and Notation (DMN) is about about rules and decisions. The three models can be used together, however, caution is necessary when modelling and linking them. Clear modelling rules and guidelines are needed to ensure consistent models that cooperate but do not obstruct each other.

When it comes to linking the models, the three OMG standards are designed to be compatible with each other: in BPMN, a case task can be used to call upon an underlying CMMN model. Similarly, both in BPMN and in CMMN, a business rule task, which we call a decision activity, can be used to invoke an underlying DMN model. Hence, processes can be filled with decisions and cases, while the cases themselves can boast decisions as well.

The Separation of Concerns (SoC) paradigm offers firm motivation for keeping multi-perspective modelling tasks, such as control flows, cases, and decision making, isolated and founded on a basis which can be used to ensure consistency. The integrated modelling and externalisation was already considered in terms of business rules [28,29]. With DMN, externalisation of decisions from processes has become a possibility, since decisions can be encapsulated in separate decision models and linked to the invoking context. Likewise, with CMMN ad hoc activity modelling can be externalised to a separate model as well. That way, the modelling endeavour becomes clearer and the models are better to assess in terms of understandability, complexity, maintainability, and scalability [3,20]. This paper aims at not only discussing when a certain modelling standard is suitable for employment, but also at clarifying how the different modelling methods should be used together in a sound setting.

4 Integration Challenges and Opportunities

In this section, we address the pairwise integration of processes, cases, and decisions by illuminating possible inconsistencies when using two modelling methods simultaneously, each representing a modelling concern of its own. Additionally, we suggest rules and guidelines on how to remedy some of these inconsistencies.

4.1 Integrating Processes and Decisions

In this subsection, possible incompatibilities that might arise between the process model and the decision model are discussed, as the goal is to identify potential incompatibilities and subsequently to alter the process to restore consistency. This issue was discussed more in detail in [3]. Here, we enumerate the possible inconsistencies identified in [3] and briefly explain what those inconsistencies pertain to.

1. **Decision Outcome Incompatibility**. In this case, not all outcomes from the decisions are included in the process model. Decisions can (re)direct the flow of the process. In an integrated process-decision model, all outcomes of the decision should be represented in the control flow if that decision redirects the process, thus ensuring a correct conclusion of the process.
2. **Intermediate Result Incompatibility**: Inconsistencies arise when subdecisions are not modelled in the process, despite the fact that the process uses the outcome of said subdecisions. Therefore, certain parts of the flow could be disturbed and render the process model inconsistent.
3. **Subdecision Inclusion Incompatibility**: Opposite to the *Intermediate Result Incompatibility*, more subdecisions than necessary can be included in the process. This inconsistency occurs when subdecisions which do not contain relevant intermediate results for the process are modelled within the process itself.
4. **Subdecision Exclusion Incompatibility**: Depending on the outcome of certain subdecisions the control flow of the process may be diverted to include additional activities, to generate exceptions or even to lead to process termination. Excluding these subdecisions that have an influence on the control flow of the process leads to process-decision inconsistency.
5. **Decision Hierarchy Incompatibility**: Another inconsistency occurs when the order of the decision activities in the process model is contradictory to the hierarchy of the decisions in the decision model. Consequently, the process cannot function correctly, as decisions are forced to enact without the prerequisite enactment of the necessary subdecisions.
6. **Input Data Incompatibility**: Decision activities also require prerequisites in order to function correctly. These prerequisites can be the outcome of certain subdecisions, as illustrated in the *Intermediate Result Incompatibility*, but also take the form of for instance user-generated input data. The inconsistency in this case occurs when the required input data is not available in a process when a certain decision task needs to be executed.
7. **Data Recency Incompatibility**: Input data used to invoke the decision model must be up to date and if necessary, this can be enforced by using timer events in the process that generate exceptions if the data is not updated within a foreseen time window.

For the sake of clarification, we will illustrate one of the process-decision incompatibilities through an example. We opt for the **Data Recency Incompatibility**. Consider Figure 3 where the process of prospect acceptance is depicted with a *Risk level* decision requiring *Financial statements* and *ID/Passport* input objects created by activity *Receive Documents*. However these documents must be up to date in order to make a sound decision. A rule might be that if the time between the start of the process and the enactment of the *Risk level* decision is longer than six months, the documents have to be resubmitted in order to ensure that the decision is taken based on up to date documents.

Incorporating this constraint can easily be done using a timer event on the *Receive documents* activity, which will trigger the activity every six months. That

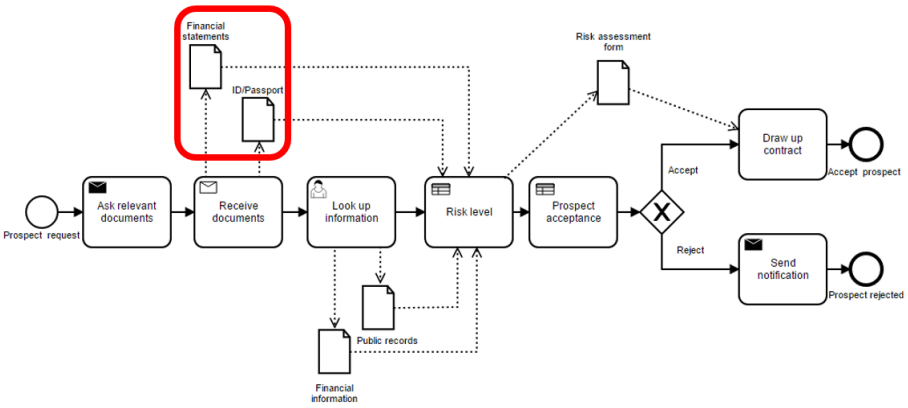

Fig. 3: Prospect acceptance process model.

way, the process will proceed with the correct data and the decision enactment will be sound. This remedies the **Data Recency Incompatibility**.

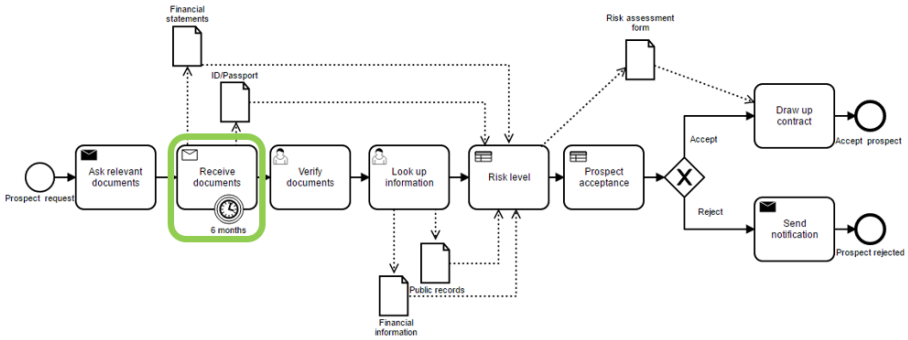

Fig. 4: Prospect acceptance process model remedied.

4.2 Integrating Processes and Cases

When using cases within the process, it is of paramount importance that the process knows the state of the case that has been executed. States and milestones that are part of the case model can be linked to paths in the process model. Hence, depending on the conclusion of the case, the process model will continue on a certain path after the enactment of the case. If there are more paths in the process than conclusions in the case model, incompatibilities between the two models can arise. We call this the **Path Reification Incompatibility**. Similarly, if the case contains conclusions that are not followed up by the control

flow of the process, the process might not be able to continue after the case enactment. We call these inconsistencies **Case Conclusion Incompatibilities**.

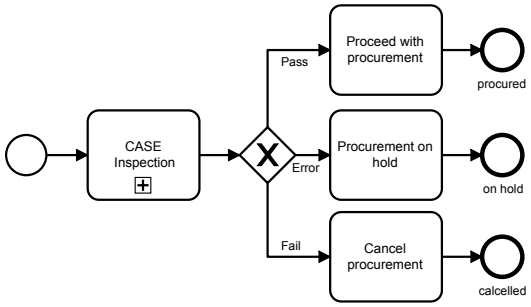

Fig. 5: Food inspection process model.

Consider the example process regarding food inspection in Figure 5. The process contains a *Case* with the name *Inspection*. Depending on the outcome of the underlying case, the process will will continue in three possible paths before reaching a conclusion. Figure 6 gives the underlying case model. However, only two possible milestones are present in the case model, while the process has three different paths leaving the gateway after the enactment of the case. Clearly, the **Path Reification Incompatibility** is at play here as discussed above.

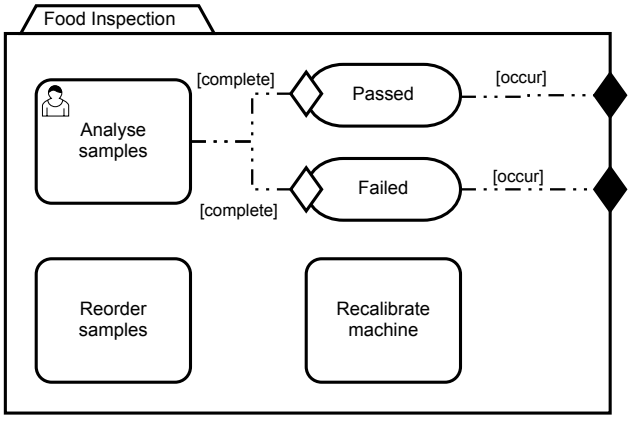

Fig. 6: Food inspection case model.

To remedy this inconsistency, either the process or the case model must be adapted. One could remove one of the paths in the process model or add an additional terminating state in the case model in the form of a milestone. We

opt for the latter and the remedied case model is presented in Figure 7. By adding the additional milestone of *Inspection error* the case and process model are rendered consistent again, and the process can continue to reach a sound conclusion. Note that a single path in the process can catch multiple terminating states from the case model through an *or* logical statement. Hence, a one-on-one mapping of terminating states and process pathways is not always necessary.

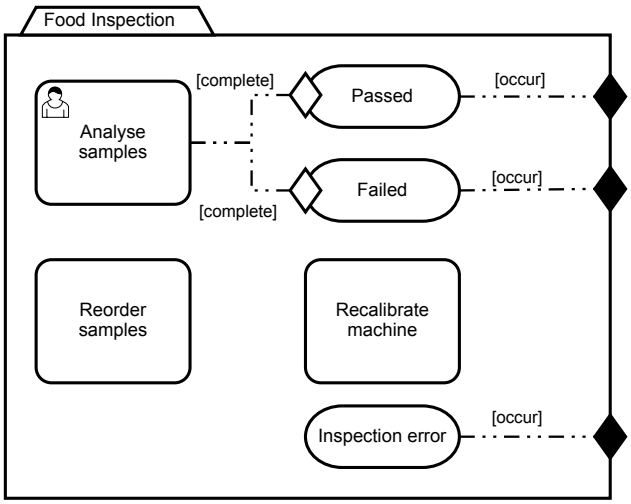

Fig. 7: Food inspection case model remedied.

4.3 Integrating Cases and Decisions

Similar to the integration of processes and decisions in Section 4.1, the **Input Data Incompatibility** can also occur when employing cases and decisions. Hence, the case model must provide the relevant input data for the decision model it wishes to invoke. Depending on the outcome of the decision, the case will reach certain milestones. Inconsistencies occur when there are more decision outcomes in the decision model than possible milestones in the process model. Thus, when certain decision outcomes are reached, the case model is not capable of detecting them. We call this the **Decision Conclusion Incompatibility**.

Consider the case model in Figure 8, depicting a contract case model. In the case, a *Check credit* decision activity is present. In order to invoke this decision correctly, the case model must provide the necessary data input for the decision enactment, otherwise the **Input Data Incompatibility** occurs. In this model, we assume the data input is provided correctly through the data object modelled within the case model. However, assume that the *Check credit* decision has three possible outcomes: *Credit OK*, *Credit Problem*, and *On Hold*. The case model in

Figure 8 only provides two milestones for the outcomes of *Credit OK* and *Credit Problem*. Hence, if the outcome *On Hold* occurs, the case model is not capable of detecting it and the **Decision Conclusion Incompatibility** manifests itself.

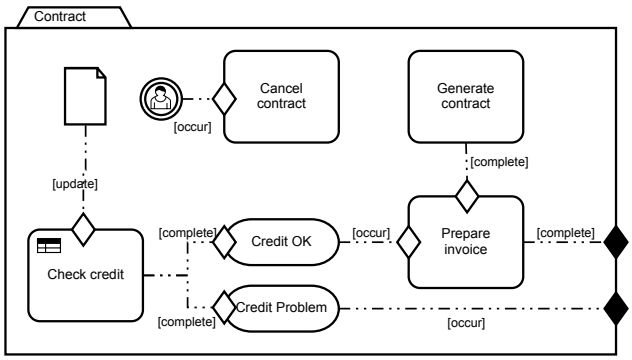

Fig. 8: Contract generation case model.

Again, there are two ways of remedying this incompatibility: either by introducing a milestone in the case model capturing the decision outcome of *On Hold* or by making the decision outcome of *On Hold* impossible to reach in the underlying decision model and thus deleting it from the decision model. We opt for the former approach and the remedied case model is presented in Figure 9. Clearly, the case model is now able to detect all outcomes of the underlying decision model and is thus capable of reaching a sound end state and conclusion.

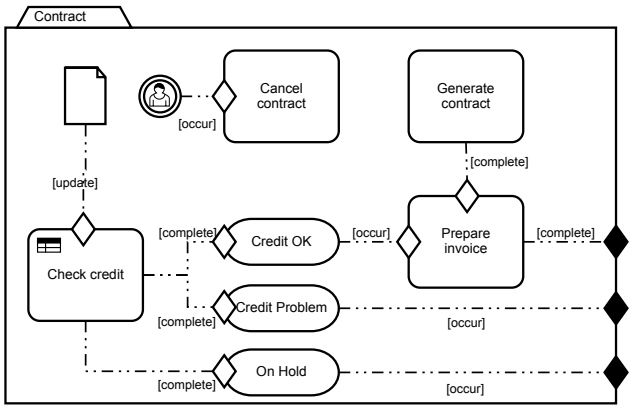

Fig. 9: Contract generation case model remedied.

5 Conclusion and Future Work

This work discusses insights and challenges for an integrated modelling approach of processes, cases, and decisions. We discussed integration scenarios and elaborated upon inconsistencies that might arise during the endeavour of model integration. Consistent integration should rely on a profound data management between the separate modelling concerns, by correctly matching intermediate results of decisions, process data, and case data necessary for the enactment of the knowledge-intensive process in its entirety.

In future work, we will further investigate how data management needs to be organised in order to reach consistency in integration of processes, cases, and decisions. Besides, the modelling complexity of integrated models is an interesting question for research as well. Additionally, integrated modelling in cooperative information systems and in distributed processes is of particular interest for Internet of Things (IoT) applications.

References

1. OMG: Decision Model and Notation 1.1 (2016)
2. Taylor, J., Fish, A., Vanthienen, J., Vincent, P.: Emerging standards in decision modeling. In: Intelligent BPM Systems: Impact and Opportunity. BPM and Workflow Handbook series, iBPMS Expo (2013) 133–146
3. Hasić, F., Devadder, L., Dochez, M., Hanot, J., De Smedt, J., Vanthienen, J.: Challenges in refactoring processes to include decision modelling. In: Business Process Management Workshops. LNBIP, Springer (2017)
4. De Smedt, J., Hasić, F., Vanthienen, J.: Towards a holistic discovery of decisions in process-aware information systems. In: Business Process Management. Lecture Notes in Business Information Processing, Springer (2017)
5. Hasić, F., De Smedt, J., Vanthienen, J.: Towards assessing the theoretical complexity of the decision model and notation (dmn). In: Enterprise, Business-Process and Information Systems Modeling. Springer International Publishing (2017)
6. OMG: Case Management Model and Notation 1.1 (2016)
7. Marin, M.A., Lotriet, H., Van Der Poll, J.A.: Measuring method complexity of the case management modeling and notation (cmmn). In: Proceedings of the Southern African Institute for Computer Scientist and Information Technologists Annual Conference 2014 on SAICSIT 2014 Empowered by Technology, ACM (2014) 209
8. Marin, M., Hull, R., Vaculín, R.: Data centric bpm and the emerging case management standard: A short survey. In: International Conference on Business Process Management, Springer (2012) 24–30
9. Marin, M.A., Hauder, M., Matthes, F.: Case management: an evaluation of existing approaches for knowledge-intensive processes. In: International Conference on Business Process Management, Springer (2015) 5–16
10. de Carvalho, R.M., Mili, H., Gonzalez-Huerta, J., Boubaker, A., Leshob, A.: Comparing condec to cmmn: Towards a common language for flexible processes. In: Model-Driven Engineering and Software Development (MODELSWARD), 2016 4th International Conference on, IEEE (2016) 233–240
11. Hinkelmann, K., Pierfranceschi, A.: Combining process modelling and case modeling. In: 8th International Conference on Methodologies, Technologies and Tools enabling e-Government MeTTeG14. (2014)

12. Hinkelmann, K.: Business process flexibility and decision-aware modelingthe knowledge work designer. In: Domain-Specific Conceptual Modeling. Springer (2016) 397–414
13. Vanthienen, J.: What business rules and tables can do for regulations. Business Rules Journal **8**(7) (2007)
14. Batoulis, K., Meyer, A., Bazhenova, E., Decker, G., Weske, M.: Extracting decision logic from process models. In: CAiSE. LNCS, Springer (2015) 349–366
15. Rao, L., Mansingh, G., Osei-Bryson, K.M.: Building ontology based knowledge maps to assist business process re-engineering. Decision Support Systems **52**(3) (2012) 577 – 589
16. Mertens, S., Gailly, F., Poels, G.: Enhancing declarative process models with dmn decision logic. In: Enterprise, Business-Process and Information Systems Modeling. Springer (2015) 151–165
17. Goedertier, S., Vanthienen, J., Caron, F.: Declarative business process modelling: principles and modelling languages. Enterprise Information Systems **9**(2) (2015) 161–185
18. Serral, E., De Smedt, J., Snoeck, M., Vanthienen, J.: Context-adaptive petri nets: Supporting adaptation for the execution context. Expert Systems with Applications **42**(23) (2015) 9307–9317
19. Weber, B., Reichert, M., Mendling, J., Reijers, H.A.: Refactoring large process model repositories. Computers in Industry **62**(5) (2011) 467–486
20. 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). (2013) 76–81
21. OMG: Business process model and notation (BPMN) 2.0 (2011)
22. Hasić, F., De Smedt, J., Vanthienen, J.: An Illustration of Five Principles for Integrated Process and Decision Modelling (5PDM). Technical report, KU Leuven (2017)
23. Hasić, F., De Smedt, J., Vanthienen, J.: A service-oriented architecture design of decision-aware information systems: Decision as a service. In: On the Move to Meaningful Internet Systems, Springer (2017)
24. Hu, J., Aghakhani, G., Hasić, F., Serral, E.: An evaluation framework for design-time context-adaptation of process modelling languages. In: Practice of Enterprise Modelling (PoEM), Springer (2017)
25. Pesic, M., Schonenberg, H., van der Aalst, W.M.: Declare: Full support for loosely-structured processes. In: Enterprise Distributed Object Computing Conference, 2007. EDOC 2007. 11th IEEE International, IEEE (2007) 287–287
26. Maggi, F.M., Dumas, M., García-Bañuelos, L., Montali, M.: Discovering data-aware declarative process models from event logs. In: BPM. Volume 8094 of Lecture Notes in Computer Science. Springer (2013) 81–96
27. Montali, M., Chesani, F., Mello, P., Maggi, F.M.: Towards data-aware constraints in declare. In: Proceedings of the 28th annual ACM symposium on applied computing, ACM (2013) 1391–1396
28. Goedertier, S., Vanthienen, J.: Compliant and flexible business processes with business rules. In: Proceedings of the CAISE Workshop on Business Process Modelling, Development, and Support BPMDS. (2006)
29. Wei, W., Indulska, M., Sadiq, S.: Guidelines for business rule modeling decisions. Journal of Computer Information Systems (2017) 1–11