Decision Support for Knowledge-Intensive Processes

Anjo Seidel and Stephan Haarmann

Hasso Plattner Institute, University of Potsdam,
Prof.-Dr.-Helmert-Str. 2-3, 14482 Potsdam, Germany
anjo.seidel@student.hpi.de, stephan.haarmann@hpi.de

Abstract. In knowledge-intensive processes, knowledge workers have to choose from many actions those that align best with their objectives. This is challenging since such a decision involves explicit and tacit knowledge and may affect the future of the process in intricate ways. In other words, they cause a high cognitive load. Using flexible case models, we present an automated recommender system that determines the best possible action for given key performance indicators. This supports knowledge workers to accomplish their goals efficiently.

Keywords: Case Management · Decision Support · Recommendations

1 Introduction

Knowledge-intensive business processes (KiPs) are characterized as multi-variant and unpredictable [2], calling for flexibility at design- and run-time [2]. Hence, new modeling approaches have emerged, which are more declarative [11, 16] and data-centric [3, 12, 13, 19] than traditional, imperative ones (e.g., such as BPMN).

With the help of an execution engine, modeled processes can be enacted [24]. At run-time, knowledge workers drive a case by deciding which of the possible next actions to execute. These decisions are interconnected and knowledge-intensive [22] and drive the process gradually towards its goal.

Due to the flexibility, knowledge workers may choose from numerous activities, and the effect of a particular activity on the process outcome is not necessarily apparent. This makes it difficult to plan the execution of KiPs, i.e., arranging actions in a sequence leading to a certain goal. Planning, however, is characteristic for knowledge work [17]. In KiPs, goals are typically defined by the knowledge workers at run-time. This is called *late goal modeling* [2].

Different approaches of providing recommendations to support planning exist, including predictive process monitoring techniques [4, 21] and decision support via process simulation [18, 25]. However, both approaches cannot be applied to KiPs, as these processes are unrepeatable and unpredictable [2].

Therefore, we propose a model-based approach for providing recommendations. In [6], we already presented a solution to allow knowledge workers to define objectives during run-time. Objectives describe desired case states. We aim to

J. Manner, D. Lübke, S. Haarmann, S. Kolb, N. Herzberg, O. Kopp (Eds.): 14th ZEUS Workshop, ZEUS 2022, Bamberg, held virtually due to Covid-19 pandemic, Germany, 24–25 February 2022, published at http://ceur-ws.org

analyze the model and the execution context to recommend how to reach such a state. Two research questions emerge:

RQ1 What are the requirements for recommendations in KiPs?
RQ2 How can such recommendations be derived?

Our approach is based on fragment-based Case Management [10]. We analyze the nature of KiPs and the requirements for late goal modeling. To provide recommendations, we query the state space of a case model and search for activities that most likely lead to desired states.

In Sect. 2, we present related work. The groundwork regarding fragment-based Case Management and modeling objectives is elaborated in Sect. 3, while our approach is elaborated in Sect. 4. We discuss the current state of work and future research and conclude the paper in Sect. 5.

2 Related Work

KiPs are highly flexible and driven by the decisions of knowledge workers [2, 20]. Various approaches for modeling knowledge-intensive processes have been proposed: some are declarative, like DECLARE [16] and Dynamic Condition Response Graphs [11]. Others are data-centric, such as Guard-Stage-Milestone [12], PHILharmonicFlows [13], and BAUML [3]. The survey papers by Di Ciccio et al. [2] and Steinau et al. [19] provide an overview of knowledge-intensive and data-centric approaches, respectively.

The limited support for data in declarative approaches and for activities in data-centric approaches, calls for *hybrid ones* [1], one of which is fragment-based Case Management [10]. This approach focuses on highly structured process fragments that can be combined dynamically during run-time. It allows combining imperative control flow and declarative data flow. Recent extensions define the modeling of data associations [7], multiplicity constraints [9], and colored Petri net semantics [5]. However, the models use implicit data flow to buy flexibility at the cost of comprehensibility, challenging knowledge workers in planning actions.

Planning is an important task in knowledge work [17]. Marella et al. proposed an approach for automating planning in business processes [14,15], which does not apply to the knowledge worker-centric nature of KiPs. Wynn et al. and Rozinat et al. provide decision support based on simulating business processes [18, 25]. As KiPs are unrepeatable and unpredictable [2], a non-repeatable simulation provides only limited support. Furthermore, predictive business process monitoring approaches aim at predicting the next actions to be executed [4, 21]. Those predictions are based on past executions, which, again, contradicts the unrepeatable and unpredictable nature of KiPs.

The challenge of assisting planning KiPs remains open. First steps have been made by providing a framework for knowledge workers to define objectives [6]. In this paper, We show how objectives can be used to derive recommendations.

3 Backgound

Our approach is based on the fragment-based case management (fCM) approach. Furthermore, this paper continues our work of allowing knowledge workers to define objectives during run-time [6]. In the following, we provide an overview of the fCM approach and our previous work regarding modeling objectives.

3.1 Fragment-Based Case Management

Fragment-based case management (fCM) combines imperative control flow and declarative data flow [10]. In fCM, the process is composed of multiple fragments, which are control flow graphs similar to BPMN models. Additionally, data flow defines data requirements and operations of activities. It constrains how fragments can be combined during run-time. An fCM case model furthermore includes a data model, object behaviors, and a termination condition. The data model consists of data classes, associations, and multiplicity constraints [5, 7, 9]. Each data class has a state transition system defining the behavior of corresponding objects. The termination condition specifies the goal of the process.

In the following, we introduce the exemplary case model for assessing and deciding on insurance claims. A more detailed explanation of the example can be found online[1].

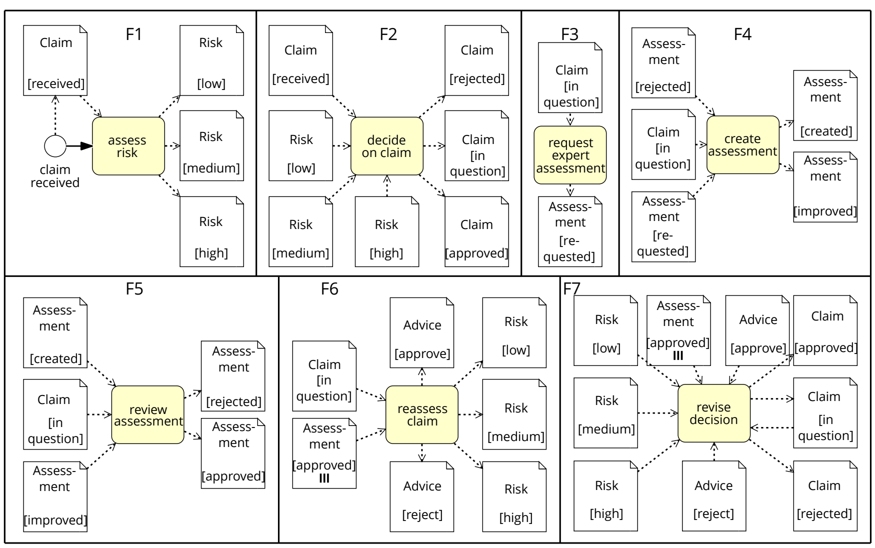

Fig. 1. Extract of fragments of the insurance claim handling process.

[1] The detailed example is available at `https://github.com/AnjoSs/DS4KiPs`

The process starts with receiving a claim. The first fragment *F1* is executed, and a risk is assessed. Given the risk, the knowledge worker can decide on the claim in *F2*. It can be accepted, rejected, or remain in question. A case in the state in question must be reassessed. During the reassessment, multiple expert assessments can be requested (*F3*), created (*F4)*, and reviewed (*F5)*. With the resulting assessments, the claim can be reassessed (*F6*), and the decision on the claim can be revised (*F7*).

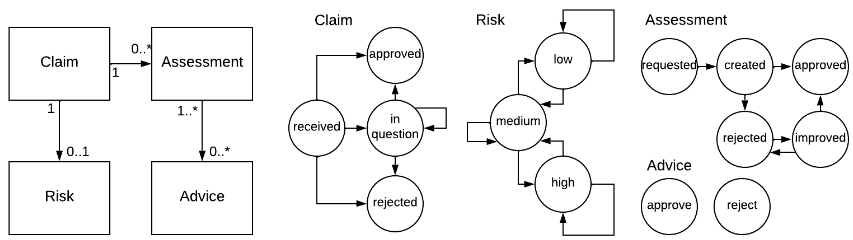

Fig. 2. The data model and object behaviors of the insurance claim handling process.

The data objects are instances of the classes *Claim*, *Risk*, *Assessment*, and *Advice* (see Fig. 2). Each claim can have one risk, and multiple expert assessments. From a number of assessments, an advice object can be retrieved. A claim can be in the states *received*, *approved*, *in question*, and *rejected*. A risk can be *low*, *medium*, or *high*. However, it cannot be changed from low to high or vice versa. An assessment can be *rejected*, *created*, then *approved* or *rejected* and *improved*. An advice can be either to *approve* or *reject* the claim

3.2 Modeling Objectives

In [6], we present a framework for specifying objectives based on an fCM model. Objectives are constraints on the state of a case. They can refer to data objects, their relationships, and to activities.

A case includes data, described by a set of data object O and a set of links L. Each object $o \in O$ belongs to a class $o.class$ and has an ID $o.id$ and a state $o.state$. A link $l \in L$ is an unordered pair of data objects.

Furthermore, each case has a set A of activity instances, henceforth called actions. An action $a \in A$ is an instance of an activity $a.activity$. It has a state $a.state$, which is either initial, control flow enabled, data flow enabled, enabled, running, or terminated [10]. Furthermore, an action reads a set of data objects $a.reads$ and writes a set of data objects $a.writes$. By executing an action, the state of the case (i.e., the sets O, L, and A) change. Using first-order logic, we can express knowledge workers' objectives using O, L, and A.

The objective g_1, for example, requires an enabled instance of activity revise decision reading an advice in state approve:

$$g_1 \equiv \exists a \in A, \exists o \in a.reads : a.activity = (revise\ decision) \wedge a.state = enabled$$
$$o.class = Advice \wedge o.state = approve$$

Multiple objectives can furthermore be composed by defining a partial order among them. It specifies the order in which the objectives need to be accomplished.

4 Recommendations for Knowledge Workers

With the opportunity to specify objectives at hand, the question is how to derive recommendations for the knowledge worker. Our approach focuses on analyzing the state space of the model itself. As the objectives are subject to the characteristics of late goal modeling, knowledge workers have special requirements for their recommendations. In the following, we elaborate on these requirements and explain how to derive suitable recommendations from a case.

4.1 Recommendation Requirements

KiPs are emergent [2]. Thus, it is impossible to plan far ahead. Instead, recommendations should focus on the immediate decision of choosing the next action. Yet, decisions still need to be made by knowledge workers, as they may have knowledge that is not part of the case state. To support workers, we calculate a score for all possible next actions. Purely based on the model, the action with the highest score aligns best with the objectives of the worker, i.e., it is recommended.

Objectives arise during run-time [9]. As the execution context may change, new objectives arise, and existing objectives change or become obsolete [2]. A knowledge worker must be able to update their objectives during run-time. Subsequently, recommendations can be calculated and actions can be (re)planned.

Weinzierl et al. [23] state that recommendations should be made w.r.t. to key performance indicators, which can be derived from data objects or past executions (i.e., event logs). In our approach, the key performance indicators are combined into a path cost function. Constant costs for all paths are equal to no cost function. Another simple implementation costs a path according to its length (number of activities). In summary, we require two user inputs:

1. A set of objectives that need to be fulfilled in the future.
2. A path cost function representing meaningful key performance indicators.

The expected results of recommendations and the described user inputs define the requirements of knowledge workers towards recommendations. **RQ1** is answered.

Consider our example from Sect. 3. The knowledge worker has specified the objective g_1 requiring *revise decision* reading an advice object in the state approve to be enabled. Assuming the case is in a state in which the claim has state *in* *question*, the risk is *medium*, two assessments are already *approved*, and no advice exists yet. The tasks *reassess claim* and *request expert assessment* are enabled. Now, a new objective g_2 emerges. It requires *revise decision* to be enabled for an advice object linked to at least three approved assignments.

Starting in the current state, the knowledge worker is interested in reaching the objectives g_1 and g_2. As a path cost function, the objectives should be reached with as few activities as possible. Therefore, we calculate a corresponding score for the next activities *reassess claim* and *request expert assessment*.

4.2 Deriving Recommendations

A business process model can be encoded into a planning domain [15], which can be used to derive recommendations. For this purpose, we reuse fCM's colored Petri net formalization [5,8]. It enables us to calculate and explore the model's state space, i.e., a directed graph consisting of all states and state transitions.

We calculate the scores for actions as follows (cf. Alg. 1): For each action, we start a breath-first search in the target state. We search for paths that result in a state satisfying the knowledge worker's objectives. For each such path, we calculate its costs. The inverse of the cost is added to the action's score. The rationale behind this scoring function is "if more cheap paths satisfying the objectives exist, the score of an action is higher." In other words, an action scores higher if it is likely to lead efficiently to a state, where all objectives are satisfied.

Algorithm 1 The score evaluation for next activities

```
function retrieve_recommendations(current_state, objectives, path_cost_function)
    action_scores ← [ ]
    Q ← queue(next(current_state))
    while Q is not empty do
        current_path ← Q.pop()
        if objectives hold in current_path[last] then
            action_scores at current_path[0] += 1 ÷ path_cost_function(current_path)
        else
            for next_action in next(current_path) do
                Q.push(current_path.append(next_action))
            end for
        end if
    end while
    return action_scores
end function
```

The presented algorithm provides a solution for deriving recommendations according to their requirements. It addresses and answers **RQ2**.

Considering the example, in the current state, *reassess claim* and *request expert assessment* are enabled. For both, a score is computed how likely they efficiently lead to a state, where g_1 and g_2 hold. All paths that start by executing *reassess claim* create an advice with only two assessments. This does not suffice to satisfy g_2. A new advice would need to be created with three or more assessments. On the other side, by executing *request expert assessment*, it is possible to create and review a new assessment, and to create the advice based on three assessments directly. There are shorter paths starting in *request expert assessment* than those starting in *reassess claim*. Therefore, Alg. 1 will rank *request expert assessment* higher than *reassess claim*.

5 Discussion and Conclusion

In our approach, we propose the use of a breadth-first search algorithm. The state space of a case grows exponentially and is possibly infinite. Search algorithms might not terminate. In combination with useful termination conditions, a breadth-first search can terminate early and lead to approximate results without querying the whole state space. The algorithm aims to find all reachable states where the objective holds, it derives optimal results for the specified path cost function. What especially suitable path cost functions look like, still needs to be evaluated.

For evaluation, we implemented a first prototype[2], which makes simple recommendations. It uses fCM's colored Petri net formalization and CPN-Tools[3] [5, 8]: By analyzing the model's state space, our prototype can verify for each possible next action whether the objectives can be satisfied eventually. This allows knowledge workers to assess whether an action complies with their objectives.

In future work, we plan to extend the prototype. First, knowledge workers need to be allowed to input the objectives and the cost function. Second, the prototype needs to calculate and return the scores of actions. Also, some technical challenges need to be addressed. Due to the flexibility of fCM, the state space is expected to grow exponentially. The algorithm for the state space search profits from optimization. The definition of fCM allows the state space even to be infinite, so the algorithm might not terminate at all. In practice, useful termination conditions for the search need to be found. Furthermore, a qualitative evaluation in the form of a user study can help to gain insights for the presented approach and prove it to work.

In this paper, we propose a framework allowing knowledge workers to state their requirements toward recommendations. These requirements consist of objectives and a path cost function, which encodes meaningful key performance indicators. The case model's state space is then analyzed in the search for paths towards states that satisfy the objectives. The more likely an action is to be part of such paths, and the cheaper the paths are, the higher the action is recommended.

With our work, we aim to support knowledge workers in making decisions. This support is a great asset for utilizing knowledge-intensive processes in practice.

[2] https://github.com/bptlab/fCM-query-generator/tree/ZEUS_2022
[3] http://cpntools.org

References

1. Andaloussi, A.A., Burattin, A., Slaats, T., Kindler, E., Weber, B.: On the declarative paradigm in hybrid business process representations: A conceptual framework and a systematic literature study (extended abstract). EMISA Forum **41**(1), 19–20 (2021)
2. Ciccio, C.D., Marrella, A., Russo, A.: Knowledge-intensive processes: Characteristics, requirements and analysis of contemporary approaches. J. Data Semant. **4**(1), 29–57 (2015), https://doi.org/10.1007/s13740-014-0038-4
3. Estañol, M., Queralt, A., Sancho, M., Teniente, E.: Artifact-centric business process models in UML. In: Rosa, M.L., Soffer, P. (eds.) Business Process Management Workshops - BPM International Workshops, Tallinn, Estonia, September 3, 2012. Revised Papers. Lecture Notes in Business Information Processing, vol. 132, pp. 292–303. Springer (2012)
4. Francescomarino, C.D., Ghidini, C., Maggi, F.M., Milani, F.: Predictive process monitoring methods: Which one suits me best? In: Weske, M., Montali, M., Weber, I., vom Brocke, J. (eds.) Business Process Management - 16th International Conference, BPM 2018, Sydney, NSW, Australia, September 9-14, 2018, Proceedings. Lecture Notes in Computer Science, vol. 11080, pp. 462–479. Springer (2018), https://doi.org/10.1007/978-3-319-98648-7_27
5. Haarmann, S., Montali, M., Weske, M.: Refining case models using cardinality constraints. In: Rosa, M.L., Sadiq, S.W., Teniente, E. (eds.) Advanced Information Systems Engineering - 33rd International Conference, CAiSE 2021, Melbourne, VIC, Australia, June 28 - July 2, 2021, Proceedings. Lecture Notes in Computer Science, vol. 12751, pp. 296–310. Springer (2021), https://doi.org/10.1007/978-3-030-79382-1_18
6. Haarmann, S., Seidel, A., Weske, M.: Modeling objectives of knowledge workers. In: Marrella, A., Weber, B. (eds.) Business Process Management Workshops - BPM 2021 International Workshops, Rome, Italy, September 6-10, 2021, Revised Selected Papers. Lecture Notes in Business Information Processing, vol. 436, pp. 337–348. Springer (2021), https://doi.org/10.1007/978-3-030-94343-1_26
7. Haarmann, S., Weske, M.: Correlating data objects in fragment-based case management. In: Abramowicz, W., Klein, G. (eds.) Business Information Systems - 23rd International Conference, BIS 2020, Colorado Springs, CO, USA, June 8-10, 2020, Proceedings. Lecture Notes in Business Information Processing, vol. 389, pp. 197–209. Springer (2020), https://doi.org/10.1007/978-3-030-53337-3_15
8. Haarmann, S., Weske, M.: Cross-case data objects in business processes: Semantics and analysis. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) Business Process Management Forum - BPM Forum 2020, Seville, Spain, September 13-18, 2020, Proceedings. Lecture Notes in Business Information Processing, vol. 392, pp. 3–17. Springer (2020), https://doi.org/10.1007/978-3-030-58638-6_1
9. Haarmann, S., Weske, M.: Data object cardinalities in flexible business processes. In: del-Río-Ortega, A., Leopold, H., Santoro, F.M. (eds.) Business Process Management Workshops - BPM International Workshops, Seville, Spain, September 13-18, 2020, Revised Selected Papers. Lecture Notes in Business Information Processing, vol. 397, pp. 380–391. Springer (2020)
10. Hewelt, M., Weske, M.: A hybrid approach for flexible case modeling and execution. In: Rosa, M.L., Loos, P., Pastor, O. (eds.) Business Process Management Forum - BPM Forum 2016, Rio de Janeiro, Brazil, September 18-22, 2016, Proceedings. Lecture Notes in Business Information Processing, vol. 260, pp. 38–54. Springer (2016), https://doi.org/10.1007/978-3-319-45468-9_3

11. Hildebrandt, T.T., Mukkamala, R.R.: Declarative event-based workflow as distributed dynamic condition response graphs. In: Honda, K., Mycroft, A. (eds.) Proceedings Third Workshop on Programming Language Approaches to Concurrency and communication-cEntric Software, PLACES 2010, Paphos, Cyprus, 21st March 2010. EPTCS, vol. 69, pp. 59–73 (2010), https://doi.org/10.4204/EPTCS.69.5
12. Hull, R., Damaggio, E., Fournier, F., Gupta, M., III, F.F.T.H., Hobson, S., Linehan, M.H., Maradugu, S., Nigam, A., Sukaviriya, P., Vaculín, R.: Introducing the guard-stage-milestone approach for specifying business entity lifecycles. In: Bravetti, M., Bultan, T. (eds.) Web Services and Formal Methods - 7th International Workshop, WS-FM 2010, Hoboken, NJ, USA, September 16-17, 2010. Revised Selected Papers. Lecture Notes in Computer Science, vol. 6551, pp. 1–24. Springer (2010), https://doi.org/10.1007/978-3-642-19589-1_1
13. Künzle, V., Reichert, M.: Philharmonicflows: towards a framework for object-aware process management. J. Softw. Maintenance Res. Pract. **23**(4), 205–244 (2011), https://doi.org/10.1002/smr.524
14. Marrella, A.: What automated planning can do for business process management. In: Teniente, E., Weidlich, M. (eds.) Business Process Management Workshops - BPM 2017 International Workshops, Barcelona, Spain, September 10-11, 2017, Revised Papers. Lecture Notes in Business Information Processing, vol. 308, pp. 7–19. Springer (2017), https://doi.org/10.1007/978-3-319-74030-0_1
15. Marrella, A.: Automated planning for business process management. J. Data Semant. **8**(2), 79–98 (2019), https://doi.org/10.1007/s13740-018-0096-0
16. Pesic, M., Schonenberg, H., van der Aalst, W.M.P.: DECLARE: full support for loosely-structured processes. In: 11th IEEE International Enterprise Distributed Object Computing Conference (EDOC 2007), 15-19 October 2007, Annapolis, Maryland, USA. pp. 287–300. IEEE Computer Society (2007), https://doi.org/10.1109/EDOC.2007.14
17. Pyöriä, P.: The concept of knowledge work revisited. J. Knowl. Manag. **9**(3), 116–127 (2005), https://doi.org/10.1108/13673270510602818
18. Rozinat, A., Wynn, M.T., van der Aalst, W.M.P., ter Hofstede, A.H.M., Fidge, C.J.: Workflow simulation for operational decision support. Data Knowl. Eng. **68**(9), 834–850 (2009), https://doi.org/10.1016/j.datak.2009.02.014
19. Steinau, S., Marrella, A., Andrews, K., Leotta, F., Mecella, M., Reichert, M.: DALEC: a framework for the systematic evaluation of data-centric approaches to process management software. Softw. Syst. Model. **18**(4), 2679–2716 (2019), https://doi.org/10.1007/s10270-018-0695-0
20. Swenson, K.D.: Position: BPMN is incompatible with ACM. In: Rosa, M.L., Soffer, P. (eds.) Business Process Management Workshops - BPM 2012 International Workshops, Tallinn, Estonia, September 3, 2012. Revised Papers. Lecture Notes in Business Information Processing, vol. 132, pp. 55–58. Springer (2012), https://doi.org/10.1007/978-3-642-36285-9_7
21. Teinemaa, I., Dumas, M., Rosa, M.L., Maggi, F.M.: Outcome-oriented predictive process monitoring: Review and benchmark. ACM Trans. Knowl. Discov. Data **13**(2), 17:1–17:57 (2019), https://doi.org/10.1145/3301300
22. Vaculín, R., Hull, R., Heath, T., Cochran, C., Nigam, A., Sukaviriya, P.: Declarative business artifact centric modeling of decision and knowledge intensive business processes. In: Proceedings of the 15th IEEE International Enterprise Distributed Object Computing Conference, EDOC 2011, Helsinki, Finland, August 29 - September 2, 2011. pp. 151–160. IEEE Computer Society (2011), https://doi.org/10.1109/EDOC.2011.36

23. Weinzierl, S., Dunzer, S., Zilker, S., Matzner, M.: Prescriptive business process monitoring for recommending next best actions. In: Fahland, D., Ghidini, C., Becker, J., Dumas, M. (eds.) Business Process Management Forum - BPM Forum 2020, Seville, Spain, September 13-18, 2020, Proceedings. Lecture Notes in Business Information Processing, vol. 392, pp. 193–209. Springer (2020), https://doi.org/10.1007/978-3-030-58638-6_12
24. Weske, M.: Business Process Management - Concepts, Languages, Architectures, Third Edition. Springer (2019), https://doi.org/10.1007/978-3-662-59432-2
25. Wynn, M.T., Dumas, M., Fidge, C.J., ter Hofstede, A.H.M., van der Aalst, W.M.P.: Business process simulation for operational decision support. In: ter Hofstede, A.H.M., Benatallah, B., Paik, H. (eds.) Business Process Management Workshops, BPM 2007 International Workshops, BPI, BPD, CBP, ProHealth, RefMod, semantics4ws, Brisbane, Australia, September 24, 2007, Revised Selected Papers. Lecture Notes in Computer Science, vol. 4928, pp. 66–77. Springer (2007), https://doi.org/10.1007/978-3-540-78238-4_8