Object-Sensitive Action Patterns in Process Model Repositories

Sergey Smirnov[1], Matthias Weidlich[1], Jan Mendling[2], and Mathias Weske[1]

[1] Hasso Plattner Institute, Potsdam, Germany
{sergey.smirnov,matthias.weidlich,mathias.weske}@hpi.uni-potsdam.de
[2] Humboldt-Universität zu Berlin, Germany
jan.mendling@wiwi.hu-berlin.de

Abstract. Organizations maintain large repositories of business process models. While maintenance and management of these repositories are challenging, they also offer opportunities when used as a knowledge base systematically. For instance, repositories can be leveraged to provide modeling support and, therefore, help to assure the consistency of newly created models with the existing ones. In the previous work we have introduced action patterns as reusable blocks of process models that can be derived from a model repository. In this paper we advance the initial results interpreting the action concept as a composition of a verb and a business object. The subsequently identified action pattern types allow for fine-grained modeling support. We evaluate the novel concepts and compare them to the established action patterns using as a benchmark the SAP Reference Model, the real world process model collection.

1 Introduction

Enterprises perceive business process management as a vehicle to achieve competitive advantage. As most process management methodologies assume availability of models formally capturing business processes, large enterprises maintain process model collections with hundreds or even thousands of process models. Maintenance of such collections brings up new challenges, among them process variants management and synchronization of several models capturing one business case. At the same time, a model collection is a valuable knowledge resource. The models describe a business domain, capturing relations between activities, events, and data objects. We believe that a thorough analysis of a model collection helps to address challenges in the context of model creation and maintenance. This approach is well-known in engineering and usually referred to as the *reuse* principle.

Various approaches that leverage reuse principles to increase process modeling efficiency have been proposed. For instance, reference modeling accumulates the domain knowledge in a reference model, which is further customized in different application projects [1]. On the opposite, several types of patterns for process models describe recurring situations in a domain independent way [2, 3]. Whilst such patterns are well-suited for model verification and generic modeling support,
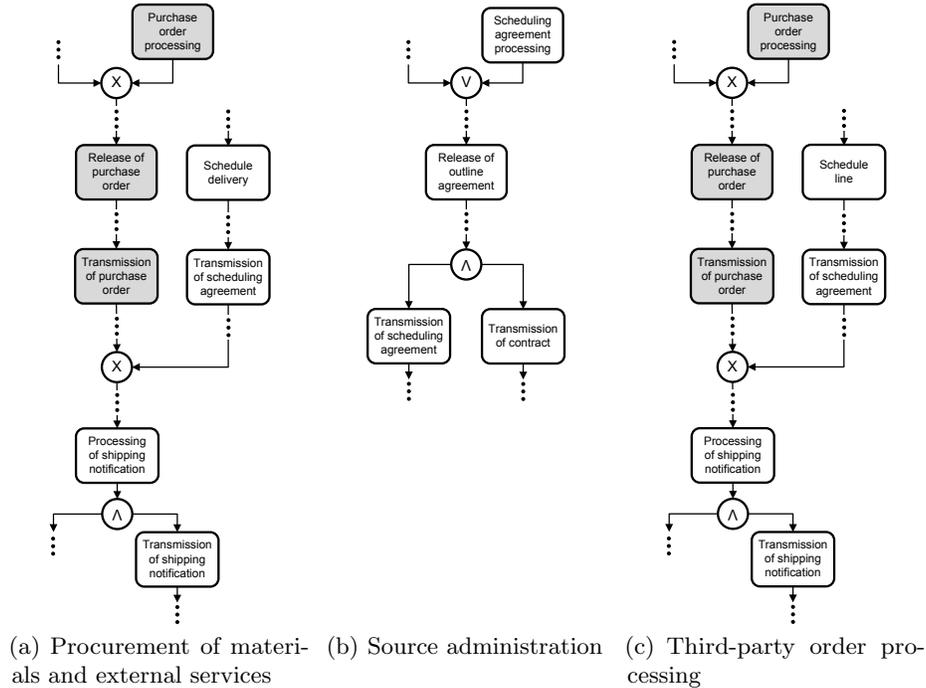
the existing reference models are tightly coupled with their partial domain and can hardly be used in other settings. Recently, several approaches emerged that aim at addressing this inherent trade-off between semantic richness of patterns and their applicability in a broad context [4, 5]. In [5], we have introduced an *action pattern* concept and described a method for action pattern mining. The term *action* essentially refers to the verb that describes the work content of an activity. *Action patterns* capture relations between actions. In contrast to workflow patterns [2], action patterns are related to the process model business semantics, yet, unlike reference models, action patterns are abstract enough to be reused in various domains. Thereby, action patterns, for example, can support the modeler during a process model design. Notice that such an interpretation of an action ignores information about the objects to which the action is applied. Actions capturing both a verb and an object provide more information about the model semantics. Hence, the use of object-sensitive actions makes the difference in the context of modeling support: more precise recommendations can be delivered to the modeler.

In this paper we elaborate on the action concept in more detail. In particular, the relations of actions and their subjects, business objects on which the actions are performed, are studied. Once business objects are taken into account, we identify different classes of actions. Hence, different classes of action patterns are derived. Our contribution is the description and formalization of object-sensitive action patterns along with methods for their mining based on association rule learning.

The remainder of the paper is structured as follows. Section 2 describes the background of our work in terms of modeling support based on action patterns. Section 3 introduces the novel concept of object-sensitive action patterns. Section 4 reports on findings on the application of our approach to a collection of industry process models, i.e., the SAP Reference Model [6]. Section 5 discusses related work, before Section 6 concludes the paper.

2 Background

The increasing amount of casual modelers imposes serious challenges for mechanisms that assure process model quality. Among numerous types of support that aim at increasing model quality we study one: a method providing recommendations on the missing activities given an incomplete model. Such recommendations are of particular value in the context of large collaborative modeling initiatives. In this setting, the risk of inconsistent process models is high due to several modelers working on similar processes. Here, domain knowledge manifested in existing process models can be leveraged to assure completeness and consistency of newly created process models. To enable the suggestions we consider the business semantics of model activities. Further, we focus on the activity perspective, as activities are the first class citizens in common process modeling languages, e.g., the Business Process Modeling Notation (BPMN) and Event-driven Process Chains (EPCs).

(a) Procurement of materials and external services (b) Source administration (c) Third-party order processing

Fig. 1. Fragments of three processes from the SAP Reference Model

Action Patterns. In order to address the need for semantic rich modeling support, we have introduced the notions of *actions* and *action patterns* in [5]. An action corresponds to the verb that describes the work content of an activity. For instance, in activity *Purchase order processing* the action is *process*. This example also illustrates the challenge of action mining: *purchase* can be interpreted as action as well. However, promising results on action mining have been presented for labels of process model activities [7]. Hence, we assume the existence of a mechanism that extracts actions from activity labels. Action patterns organize domain specific knowledge in terms of actions and their relations. The goal of action patterns is to define which actions often occur together in processes and which ordering constraints exist between these actions.

Let us consider a motivating example that appears in the SAP Reference Model [6]. Fig. 1 depicts the fragments of three EPCs from this model collection. All the models describe business processes from the procurement domain. A rough inspection of the models reveals that while model fragments in Fig. 1(a) and Fig. 1(c) are extremely similar, the fragment in Fig. 1(b) is different. Indeed, fragments in Fig. 1(a) and Fig. 1(c) contain almost identical sets of activities, and, hence, actions: *process, release*, and *transmit*. Although the fragment in Fig. 1(b) contains other activities, the set of actions is the same. Thereby, the three discussed fragments are close in terms of observed actions.

Types of Action Patterns. We distinguish two types of action patterns. *Co-occurrence action patterns* capture sets of actions, which often occur together in business processes. Given a set of actions, a co-occurrence pattern specifies which actions are expected to appear as well. An example derived from the process models depicted in Fig. 1 is {*process, release*} $\Rightarrow$ {*transmit*}. This pattern suggests that once actions *process* and *release* appear in the business process, action *transmit* should also be observed. Co-occurrence action patterns may facilitate model design by suggesting actions missing in the model. In other words, these patterns propose actions to be added to the process.

Co-occurrence action patterns do not reflect the ordering relations between actions. *Behavioral action patterns* address this issue, pointing to actions and the ordering constraints between them. For a given set of actions and relations between a subset of these actions, a behavioral pattern specifies the missing behavioral relations. To this end, we use behavioral profiles [8] as a behavioral abstraction. Such a profile describes behavioral relations on the level of activity pairs. A behavioral profile consists of three relations that partition the Cartesian product of all model activities, such that two activities are either in strict order, exclusive to each other, or in interleaving order. For the model in Fig. 1(a), activities *Release of purchase order* and *Transmission of purchase order* are in strict order, as the former is executed before the latter in any case. Activities *Transmission of scheduling agreement* and *Transmission of contract*, in turn, are in interleaving order as the former might be executed before the latter and vice versa. An example of a behavioral action pattern is the statement that if actions *process*, *release*, and *transmit* appear together and *release* is executed after *process*, then *transmit* is executed after *process* and after *release*. From a user perspective behavioral action patterns complement co-occurrence patterns: while a co-occurrence pattern hints on which actions are missing, a behavioral pattern suggests how to introduce these actions in the model.

Action Pattern Mining. [5] shows how association rule learning techniques can be applied to derive action patterns given a collection of process models. Association rule learning was introduced by Agrawal et al. in [9]. We shortly recall the basic principles. Let $\mathcal{I}$ be a set of items and C a collection of transactions, where each transaction T is a set of items, i.e., $T \subseteq \mathcal{I}$. Given a set of items $X \subseteq \mathcal{I}$, we say that transaction T satisfies X, if $X \subseteq T$. An association rule in a collection C is an implication of the form $X \Rightarrow Y$, where $X \cap Y = \emptyset$ and $X, Y \subset \mathcal{I}$. Based thereon, two elementary notions can be defined, i.e., *support* and *confidence*. A set $X \subseteq \mathcal{I}$ has support n in a collection C, denoted by $supp(X)$, if n transactions satisfy set X. Support can be related to statistical significance. We are interested in sets with high support and refer to a set as being *large*, if $supp(X) \geq supp_{min}$ for a given threshold $supp_{min}$. An association rule $X \Rightarrow Y$ holds in transaction collection C with confidence $c = \frac{supp(X \cup Y)}{supp(X)}$, if at least c share of transactions satisfying X, satisfies Y as well. The confidence for a rule $X \Rightarrow Y$ is denoted by $conf(X \Rightarrow Y)$. A rule's confidence reflects its strength. Again, we are interested in rules with high confidence, i.e., those that show a confidence that is higher than a threshold $conf_{min}$.

Association rule learning enables identification of action patterns as follows. For co-occurrence action patterns, we interpret actions as items and process models as transactions. Hence, a model collection is a collection of transactions. A process model satisfies an action set, if the model comprises activities that relate to all of the actions. A co-occurrence action pattern is defined as an association rule on the domain of actions associated with values for minimal support and confidence. For the case of behavioral action patterns, we first lift the relation of the behavioral profile from the level of activities to actions. Based thereon, behavioral relations of actions are interpreted as items. Thus, a behavioral action pattern is an association rule on the domain of action pairs along with their behavioral relation, which is associated with values for minimal support and confidence.

3 Object-Sensitive Action Patterns

This section introduces object-sensitive action patterns. First, Section 3.1 elaborates on the limitations of interpreting an action as being a verb only. Subsequently, Section 3.2 and Section 3.3 introduce two novel notions of action sets that underlie object-sensitive action patterns.

3.1 Action Notion Revisited

Action patterns introduced in [5] are independent of the objects on which the actions are performed. We refer to such action patterns as *object-neutral action patterns*. Arguably, two actions such as *accept* and *reject* often occur together and are executed exclusively. However, for other action combinations, patterns might solely be observed, if the notion of an action is refined by taking the respective objects into account.

For instance, in the example in Fig. 1, we observe that all models contain the actions *process* and *release*. However, the actions relate to different objects even within one model, i.e., a *purchase order*, a *shipping notification*, and a *scheduling agreement*. While this does not impact on the co-occurrence of these actions at least in our example, we observe that the three models show different behavioral relations between these actions. In Fig. 1(a) and Fig. 1(c) the action *release* (of a purchase order) can be preceded and followed by the action *process* (of a purchase order, or of a shipping notification). In contrast, the model in Fig. 1(b) shows a strict order of actions: *process* is followed by *release*.

We see that the existing notion of object-neutral action patterns aims at deriving patterns at a coarse-grained level. While object-neutral action patterns are useful for modeling support, certain patterns might solely be observed once a more fine-grained approach is taken. Such fine-grained patterns should consider the combination of a verb and the respective object as the underlying notion of an action. We refer to these patterns as *object-sensitive action patterns*.

The definition of object-sensitive action patterns leads to a different domain for an action pattern, but does not impact on the action pattern types (i.e.,

co-occurrence and behavioral action patterns) and the way they are mined using the approach summarized in Section 2. Therefore, we focus on the definition of the domain for object-sensitive action patterns and refer the reader to [5] for a formal description of the pattern mining method.

First, we postulate Γ—an alphabet of activity labels in a process model collection. Further, we assume means to extract verbs and business objects from activity labels.

Definition 1 (Verb and Business Object Function). For a given alphabet of activity labels Γ, the *verb function* $v : \Gamma \mapsto \mathcal{V}$ derives a verb from an activity label. The *business object function* $b : \Gamma \mapsto \mathcal{B}$ derives a business object from an activity label. As a shorthand notation, we use $V_\Gamma = \bigcup_{\gamma \in \Gamma} \{v(\gamma)\}$ and $B_\Gamma = \bigcup_{\gamma \in \Gamma} \{b(\gamma)\}$ to refer to the verbs and business objects of all activity labels.

For instance, if an activity is labeled *Schedule delivery*, *v(Schedule delivery) = schedule* and *b(Schedule delivery) = delivery*.

Apparently, object-neutral action patterns are build solely from the set V_Γ. That is, verbs are the domain for these patterns and represent the items in the sense of association rule learning, cf., Section 2.

3.2 Multi-Object Action Patterns

The first kind of object-sensitive action patterns builds on the notion of actions in the sense of an operation expressed by a verb and applied to a business object. Actions become tuples of verbs and business objects, and every such tuple—an item in the sense of association rule learning.

Definition 2 (Multi-Object Action Set). Let Γ be a set of activity labels. The *multi-object action set* $\mathcal{A}_O \subseteq V_\Gamma \times B_\Gamma$ contains all pairs of verbs and objects (x, y), such that $v(\gamma) = x$ and $b(\gamma) = y$ for some activity label $\gamma \in \Gamma$.

We speak of multi-object action sets, as the actions can relate to different business objects. For instance, *(process, purchase order)*, *(release, purchase order)*, and *(transmit, scheduling agreement)* would be actions derived from the exemplary process models in Fig. 1 that can be part of a multi-object action pattern. An example for a co-occurrence action pattern is $\{$*(process, purchase order), (release, purchase order)*$\} \Rightarrow \{$*(transmit, scheduling agreement)*$\}$, i.e., the observation of the actions *(process, purchase order)* and *(release, purchase order)* suggests that the action *(transmit, scheduling agreement)* should be observed as well. Similarly, behavioral action patterns can be specified, e.g., if actions *(process, purchase order)* and *(release, purchase order)* are observed in a strict order, action *(transmit, scheduling agreement)* should be observed exclusively to both.

Multi-object action patterns are more fine-grained than object-neutral action patterns. We assume that multi-object actions allow for unveiling patterns that cannot be detected when considering solely the verbs of model element labels. On the other hand, identification of object-neutral action patterns should be prioritized as several multi-object action patterns together might represent an object-neutral action pattern.

3.3 Single-Object Action Patterns

Single-object action patterns are composed of actions where verbs are applied to a *single* business object. Again, tuples of verbs and business objects are the items in the sense of association rule learning. However, in contrast to multi-object action patterns discussed in the previous section, all actions of a pattern relate to one dedicated business object. Therefore, we need the following definition of an action.

Definition 3 (Single-Object Action Set). Let Γ be a set of activity labels. For each object $o \in B_\Gamma$, the *single-object action set* $\mathcal{A}_O^o \subseteq V_\Gamma \times B_\Gamma$ contains all pairs (x, o), such that $v(\gamma) = x$ for some activity label $\gamma \in \{\omega \in \Gamma \mid b(\omega) = o\}$.

Regarding the models in Fig. 1, again, the actions *(process, purchase order)* and *(transmit, scheduling agreement)* are derived. However, these actions refer to different business objects and, therefore, cannot occur together in a single-object action pattern. Still, we see that the models in Fig. 1 contain various actions that refer to a business object, the *purchase order* object. All these actions might be used as building blocks for single-object action patterns.

4 Evaluation based on the SAP Reference Model

This section empirically evaluates the impact of object-sensitive actions on the action patterns. The section presents the results of action patterns mining in a large process model collection. We analyze the mining results comparing the patterns discovered for object-neutral and object-sensitive actions. The analysis considers both co-occurrence and behavioral action patterns.

In the evaluation, we use the SAP Reference Model [6]—a process model collection used as a benchmark for object-neutral action patterns evaluation in [5]. The SAP Reference Model includes 604 EPCs, describing business processes supported by the SAP R/3 software. The collection is organized in 29 functional branches of an enterprise, e.g., sales and accounting. The experiment evaluating co-occurrence action patterns makes use of all 604 models. The evaluation of behavioral action patterns exploits 421 models. The decrease in the model number is due to the exclusion of models with ambiguous instantiation semantics, see [10], or behavioral anomalies, see [11]. In the experiment we have used a manual mapping of activity labels to verbs and business objects. However, as discussed before, [7] shows the potential for an automation of this step.

In the first part of the experiment, we compare co-occurrence action patterns describing object-neutral and object-sensitive actions. Table 1 presents the number of action patterns for object-neutral actions. The number of patterns dramatically decreases with the growth of support and confidence values (horizontal and vertical directions in the table, respectively). Table 2 describes the observed results for co-occurrence action patterns capturing object-sensitive actions. Table 2(a) captures the results for multi-object action sets, while Table 2(b)—for single-object action sets. The number of patterns for multi-object action sets is the

Table 1. Dependency of co-occurrence pattern number for object-neutral actions in the SAP Reference Model on $conf_{min}$ and $supp_{min}$

$conf_{min}$ \ sup_{min}	3	4	5	6	7	8	9	10
0.50	7395	2353	680	563	41	29	17	11
0.60	6123	2089	610	504	33	22	12	8
0.70	5569	1535	563	469	20	12	6	6
0.80	4684	1238	501	417	15	10	5	5
0.90	4603	1157	420	377	7	3	2	2

Table 2. Dependency of co-occurrence pattern number for object-sensitive actions in the SAP Reference Model on $conf_{min}$ and $supp_{min}$

(a) Multi-object actions

$conf_{min}$ \ sup_{min}	3	4	5	6	7	8	9	10
0.50	32072	21019	6171	6010	20	15	7	1
0.60	30127	20373	6154	6007	20	15	7	1
0.70	23884	14130	6013	5870	19	14	6	1
0.80	20601	13731	6009	5867	17	14	6	1
0.90	20300	13430	5708	5588	9	6	4	0

(b) Single-object actions

$conf_{min}$ \ sup_{min}	3	4	5	6	7	8	9	10
0.50	408	119	67	53	10	5	5	1
0.60	382	105	60	50	10	5	5	1
0.70	364	87	56	46	9	4	4	1
0.80	326	73	52	43	7	4	4	1
0.90	310	57	36	28	5	2	2	0

highest among the three compared tables. This can be explained by the fact that in the SAP Reference Model one action pattern consisting of object-neutral actions is "split" into several patterns capturing object-sensitive actions. Consider an example object-neutral action set {*transmit, process, release*}. In the case of object-sensitive actions this set is split into 22 action sets, including:

Example 1 {*transmit order, process order, release order*}
Example 2 {*transmit agreement, process notification, release order*}
Example 3 {*transmit order, transmit notification, process order, release order*}

These examples cover different types of object-sensitive actions. Example 1 illustrates a set of single-object actions. Example 2—a set of multi-object actions performed on objects *order* and *agreement*. Finally, Example 3 shows that there are multi-object action sets, where one verb is performed on more than one object, e.g., *transmit agreement* and *transmit notification*. Due to this "split" of actions, the number of patterns describing object-sensitive actions can be greater than the number of patterns for object-neutral actions. Meanwhile, the support value of each individual pattern capturing object-sensitive actions is lower than the support for a pattern for object-neutral actions: the support of one generic pattern is distributed among several more specific patterns. Among the three considered action types, the number of patterns for single-object actions is the lowest. Tables 1 and 2 show that the more specific are the patterns, the bigger is the share of patterns with high confidence.

Table 3. Dependency of behavioral action pattern number for object-neutral actions *transmit*, *process*, and *release* in the SAP Reference Model on $conf_{min}$ and $supp_{min}$

$conf_{min}$ \ sup_{min}	2	3	4	5	6	7
0.50	1559	1024	602	310	50	0
0.60	1267	813	602	310	50	0
0.70	959	748	537	310	50	0
0.80	797	586	375	310	50	0
0.90	667	456	245	180	50	0

Table 4. Dependency of behavioral pattern number for a particular object-sensitive action set in the SAP Reference Model on $conf_{min}$ and $supp_{min}$

(a) {*transmit agreement, process order, release order*}

$conf_{min}$ \ sup_{min}	2	3	...	8	9
0.50	12	12	...	12	0
0.60	12	12	...	12	0
0.70	12	12	...	12	0
0.80	12	12	...	12	0
0.90	12	12	...	12	0

(b) {*transmit order, process order, release order*}

$conf_{min}$ \ sup_{min}	2	3	...	5	6
0.50	12	12	...	12	0
0.60	12	12	...	12	0
0.70	12	12	...	12	0
0.80	12	12	...	12	0
0.90	12	12	...	12	0

Behavioral action patterns describe behavioral relations between actions in large action sets. We illustrate the discussion of behavioral action patterns by the example object-neutral action set {*transmit, process, release*}. Table 3 shows the number of behavioral patterns for the set of object-neutral actions. Table 4 presents the number of object-sensitive action patterns: Table 4(a) illustrates the number of patterns for multi-object action set {*transmit agreement, process order, release order*}, while Table 4(b)—for single-object action set. Note that we consider the patterns for a particular set of object-sensitive actions. Hence, the largest number of patterns is observed for object-neutral actions. Object-sensitive actions result in a lower number of patterns, but without dropping in confidence.

The results obtained through the analysis of the SAP Reference Model can be applied as follows. Consider that a modeler designs a model for a procurement process. The current state of the model is captured in Fig. 2 by the elements colored in black. Given the action patterns captured in Tables 2 and 4, we suggest the user to insert an action *transmit* to be performed on object *agreement* into the model. This suggestion is visualized in Fig. 2 with elements colored in gray. Notice that behavioral action patterns enable us to suggest the user *how* to introduce the new activity in the model.

According to the presented evaluation results, we conclude that in the SAP Reference Model action patterns based on single-object action sets provide the most precise information, have low support, but deliver high confidence. Object-neutral action patterns are on the opposite side: they provide very generic information, but with high statistical significance. Multi-object patterns are the compromise between these two classes, balancing between the significance and the pattern strength.
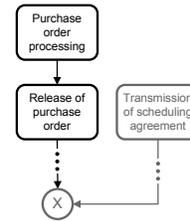
Fig. 2. A suggestion based on action patterns in Table 4

5 Related Work

First and foremost, our work relates to various *patterns* proposed for business processes. On the technical level, the workflow pattern initiative has identified patterns for several model aspects, among them the control flow [2] and the data flow [12]. On the conceptual level, Lonchamp proposed a set of collaboration patterns defining abstract building blocks for recurrent situations [13]. Tran et al. introduces a meta-model for process patterns and shows their application in the UML context [14]. Most closely related to our work is the research by Thom et al. [4]. In their work, the authors identify *workflow activity patterns* (WAP) that specify seven different types of micro workflows, e.g., approval or decision. Our action pattern approach builds on the same observation that certain activities often occur jointly to achieve an over-arching goal. In contrast to [4], we do not assume a priori knowledge on which patterns might occur in a process. Instead, object-sensitive action patterns are mined in a collection of process models. Although [15] also advocates the application of association rule learning techniques for WAP, their focus is on mining co-occurrences of these predefined patterns instead of the patterns themselves.

Recently, *intelligent support and recommendations* for process modeling has received much attention. To this end, similar models in a process model repository might be proposed as extensions to the currently modeled process using search techniques [16]. While this approach also builds on a match of actions, business objects, and textual content, we believe that action patterns are more flexible, as they do not require the knowledge about an exact continuation of a process. Consistency between object life cycles and process models is discussed in [17] along with corresponding modeling support. Moreover, modeling support might also be driven by modelers in a collaborative modeling effort [18]. In contrast to our work, this approach builds on suggestions by other modelers. Control flow correctness issues are addressed in [19] where the authors offer continuous verification of process models during modeling. In [20] the authors study how cooperative modeling is supported by fragment-driven modeling approach. However, the derivation of fragments (or action patterns) is not detailed. In order to accelerate business process modeling, structural control flow patterns can be used as suggested in [3]. Still, these suggestions do not consider the business semantics

of process models. The change patterns introduced in [21] can also be seen as patterns that allow for intelligent modeling support.

Finally, research on activity labels relates to our work. Textual labels are used for matching and comparing process models [16, 22]. Recent works by Becker et al. reuse parsing techniques from computer linguistics to identify the various parts of an activity label [23]. For the experiment of this paper, we have derived the actions manually. Still, the techniques proposed in [7, 23] have the potential to automate this step.

6 Conclusion

In this paper, we proposed object-sensitive action patterns as a means for domain-specific modeling support. These patterns are derived from a collection of process models using association rule learning techniques and can be leveraged in order to provide suggestions in the course of modeling. Co-occurrence action patterns hint at missing activities, whereas behavioral action patterns provide information on how an activity should be added to an incomplete model. Taking the relation between verbs and business objects into account, we extended the existing notion of action patterns by multi-object and single-object action patterns. Besides their formal definition, we provide an experimental evaluation of these object-sensitive patterns based on the SAP reference model. It reveals that object-sensitive patterns can be assumed to provide fine-grained modeling support.

As now action patterns have different levels of granularity, the use of synonyms in the activity labels "blurs" the patterns. In the future work, we plan to address the linguistic relations between verbs and business objects, e.g., synonymy and hyponymy. To this end, the application of thesauri like WordNet[1] might prove useful. As single-object action patterns implicitly capture the life cycle of a business object, mining of business object life cycles using action patterns in another direction of the future work. Finally, the mined action patterns call for validation proving their usefulness.

References

1. Rosemann, M., van der Aalst, W.M.P.: A Configurable Reference Modelling Language. IS **32**(1) (2007) 1–23
2. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. DPD **14**(1) (2003) 5–51
3. Gschwind, T., Koehler, J., Wong, J.: Applying Patterns during Business Process Modeling. In: BPM 2008, Berlin, Heidelberg, Springer (2008) 4–19
4. Thom, L.H., Reichert, M., Iochpe, C.: Activity Patterns in Process-aware Information Systems: Basic Concepts and Empirical Evidence. IJBPIM **4**(2) (2009) 93–110

---

[1] `http://wordnet.princeton.edu/`

5. Smirnov, S., Weidlich, M., Mendling, J., Weske, M.: Action Patterns in Business Process Models. In: ICSOC 2009. Volume 5900 of LNCS. (2009) 115–129
6. Keller, G., Teufel, T.: SAP(R) R/3 Process Oriented Implementation: Iterative Process Prototyping. Addison-Wesley (1998)
7. Leopold, H., Smirnov, S., Mendling, J.: Refactoring of Process Model Activity Labels. In: NLDB 2010. Volume 6177 of LNCS., Springer (2010) 268–276
8. Weidlich, M., Mendling, J., Weske, M.: Efficient Consistency Measurement based on Behavioural Profiles of Process Models. IEEE TSE (2010) To appear.
9. Agrawal, R., Imielinski, T., Swami, A.N.: Mining Association Rules between Sets of Items in Large Databases. In: COMAD 1993, Washington, D.C. (1993) 207–216
10. Decker, G., Mendling, J.: Instantiation Semantics for Process Models. In: BPM 2008. Volume 5240 of LNCS., Springer (2008) 164–179
11. Mendling, J.: Metrics for Process Models: Empirical Foundations of Verification, Error Prediction, and Guidelines for Correctness. Volume 6 of LNBIP. Springer (2008)
12. Russell, N., ter Hofstede, A.H.M., Edmond, D., van der Aalst, W.M.P.: Workflow Data Patterns. Technical Report FIT-TR-2004-01, Queensland University of Technology (2004)
13. Lonchamp, J.: Process Model Patterns for Collaborative Work. In: Telecoop. (1998)
14. Tran, H.N., Coulette, B., Dong, B.T.: Broadening the Use of Process Patterns for Modeling Processes. In: SEKE, Knowledge Systems Institute Graduate School (July 2007) 57–62
15. Lau, J.M., Iochpe, C., Thom, L., Reichert, M.: Discovery and Analysis of Activity Pattern Cooccurrences in Business Process Models. In: ICEIS 2009, Springer (May 2009) 83–88
16. Hornung, T., Koschmider, A., Lausen, G.: Recommendation Based Process Modeling Support: Method and User Experience. In: ER 2008. Volume 5231 of LNCS., Springer (October 2008) 265–278
17. Küster, J.M., Ryndina, K., Gall, H.: Generation of Business Process Models for Object Life Cycle Compliance. In: BPM 2007. Volume 4714 of LNCS., Brisbane, Australia, Springer (September 2007) 165–181
18. Koschmider, A., Song, M., Reijers, H.A.: Advanced Social Features in a Recommendation System for Process Modeling. In: BIS 2009. Volume 21 of LNBIP., Poznan, Poland, Springer (April 2009) 109–120
19. Kühne, S., Kern, H., Gruhn, V., Laue, R.: Business Process Modelling with Continuous Validation. In: MDE4BPM. (September 2008) 37–48
20. Kim, K.H., Won, J.K., Kim, C.M.: A Fragment-Driven Process Modeling Methodology. In: ICCSA. Volume 3482 of LNCS., Springer (2005) 817–826
21. Weber, B., Reichert, M., Rinderle-Ma, S.: Change Patterns and Change Support Features - Enhancing Flexibility in Process-aware Information Systems. DKE **66**(3) (2008) 438–466
22. van Dongen, B., Dijkman, R., Mendling, J.: Measuring Similarity between Business Process Models. In: CAiSE 2008, Berlin, Heidelberg, Springer (2008) 450–464
23. Becker, J., Delfmann, P., Herwig, S., Lis, L., Stein, A.: Towards Increased Comparability of Conceptual Models - Enforcing Naming Conventions through Domain Thesauri and Linguistic Grammars. In: ECIS 2009. (June 2009)