Workflow Patterns for Business Process Modeling

Lucinéia Heloisa Thom[1], Cirano Iochpe[1,2], Manfred Reichert[3]

[1] Federal University of Rio Grande do Sul, Informatics Institute,
Av. Bento Gonçalves, 9500,
91501-970 Porto Alegre, Brazil
[2] Data Processing Company of Porto Alegre – PROCEMPA
Av. Ipiranga, 1200,
90.160-091 Porto Alegre, Brazil
[3]Faculty of Electrical Engineering, Mathematics and Computer Science,
University of Twente, 7500 AE, Enschede, The Netherlands
{lucineia, ciochpe}@inf.ufrgs.br
M.U.Reichert@ewi.utwente.nl

Abstract. For its reuse advantages, workflow patterns (e.g., control flow patterns, data patterns, resource patterns) are increasingly attracting the interest of both researchers and vendors. Frequently, business process or workflow models can be assembeled out of a set of recurrent process fragments (or recurrent business functions), each of them having generic semantics that can be described as a pattern. To our best knowledge, so far, there has been no (empirical) work evidencing the existence of such recurrent patterns in real workflow applications. Thus, in this paper we elaborate the frequency with which certain patterns occur in practice. Furthermore, we investigate completeness of workflow patterns (based on recurrent functions) with respect to their ability to capture a large variety of business processes.

1 Introduction

For (computerized) business processes there exists a variety of fragments which can be understood as self-contained activity blocks with a well-defined semantics (Thom, 2006). In particular, a certain *process fragment* (or *recurrent business function*) may occur several times within one (or different) process definition(s). As an example consider the evaluation process for price adjustment as depicted in Figure 1. This process comprises the following activities: (a) a decision activity (to fix whether the input is a shopping order or not); (b) activity 'send e-mail to manager informing about price adjustment'; (c) activity 'evaluate request of price adjustment'; (d) activity 'notify managers about conclusion of evaluation'; (e) activity 'notify managers about automatic approval'; and (f) activity 'prepare request to be sent'. Altogether this process comprises fragments having generic semantics that can be described as a pattern such as decision (activity *a*), notification (activities *b*, *d* and *e*), and task execution request (activities *c* and *f*). In early work we proposed a set of nine workflow patterns based on respective fragments (see Thom, 2006). In recent work we dealt with the question how to provide adequate tool support for the modeling of processes that include recurrent business functions (Thom, 2007). In this paper we present examples of recurrent workflow patterns. Taking the results from a case study we show that these patterns do not only exist in real workflow applications, but are also necessary as well as sufficient to model a large variety of workflows. We believe that the use of such patterns can improve the performance of the modeling phase significantly (since it can be based on the reuse of generic activity templates).

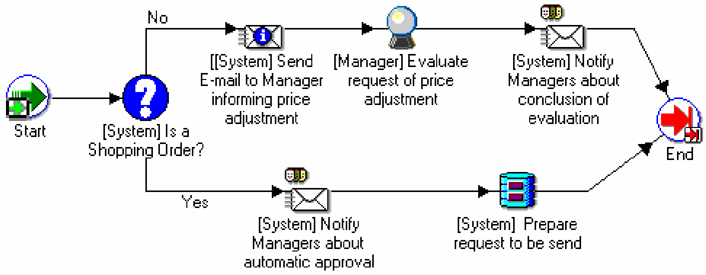

Figure 1: Evaluation process for price adjustment

So far, workflow patterns have been suggested for representing different workflow aspects: control flow (Aalst, 2002), resources (Russell, 2004), data (Russell, 2005), interactions (Bradshaw, 2005), and exception handling (Russell, 2006). All these pattern sets have in common that they are very relevant for implementing business process management (BPM) tools. Furthermore, they can be utilized for evaluating existing BPM tools and languages. However, these structural patterns provide only a partial answer to the question what business functions a modeler has to consider repeatedly in various process models.

Usually, such *process fragments* (Flores, 1988), (Medina-Mora, 1992), (Malone, 2004), (Muehlen, 2002), (Bradshaw, 2005) are re-designed for each workflow application. This lack of reusing model fragments and process knowledge has resulted in high costs and error rates regarding the modeling and maintenance of workflow-based applications. While there has been some research on how metadata can be organized to manage large-scale modeling projects (see Thomas and Scheer 2006), to our best knowledge there exists no (empirical) work evidencing the existence of recurrent patterns in real workflow applications. Furthermore, there is no work on which patterns are needed and how good they may support the modeling of whatever business process. Beyond that, contemporary workflow modeling tools do not provide functionalities that enable users to define, query, and reuse such patterns in a proper way. In (Thom, 2007) we have presented a first approach towards the implementation of our patterns based on an Event-driven Process Chains (EPCs) tool (Keller, 1992).

The remainder of this paper is structured as follows. In Section 2 we present examples of workflow patterns based on recurrent functions. In Section 3 we summarize results from an extensive case study in order to evidence the existence of such patterns as well as their completeness for workflow modeling. Finally, Section 4 concludes the paper and gives an outlook on future research.

2 Workflow Patterns

In the context of this paper we use the term workflow pattern to refer to the description of a recurrent business function frequently found in business processes (e.g., notification, decision, approval). We derived a set of 9 patterns based on an extensive literature study. Some of the patterns can be defined as specializations of other ones. We classified the patterns in 3 categories. The classification was based on specific characteristics of the processes (e.g., dependency of either application domain or organizational structure aspects).

1. *Workflow patterns based on organizational structural aspects:* This category refers to those patterns that are related to one or more organizational aspects. Examples include document *approval* and *question-answering*.
2. *Workflow patterns based on a specific application domain.* This category includes patterns that are related to a specific application domain. *Financial patterns* and *logistic patterns* are examples of this category.
3. *Workflow patterns based on recurrent functions.* This category comprises patterns related to general recurrent business functions, i.e., any kind of workflows may contain patterns of this category independent of the application domain. Examples include the *uni-* and *bi-directional performative pattern*, *informative pattern*, *notification pattern*, and *decision pattern.*

A block activity is suitable to represent each pattern according to WfMC (2005). The block activity concept is particularly suited because it allows to encapsulate the well-defined semantics and to represent their atomic characteristics. This means that all activities defined inside a block activity pattern must be completed before the superordinated workflow can continue its execution.

Block execution starts with the first activity of the block, which has no incoming transition, and continues with the other sub-activities according to their partial order. Block execution will be completed when an exit activity is reached. After block completion, the execution of the superordinated workflow continues with the activities directly succeeding this block.

Since the patterns representation may require input/output parameters and the block activity concept does not support parameters (i.e., parameters are defined in the surrounding workflow definition), the transaction perspective of serialization theory was applied to overcome this limitation (Bernstein, 1987). Accordingly, an input parameter is represented as a database read operation of *one-time-only* readable information. Similarly, an output parameter is represented in the block as a database write operation of *one-time-only* writable information.

We describe two pattern examples based on UML Activity Diagram (using the UML 2.0 notation, see Figure 2) (see Thom, 2006 for a description of all patterns).

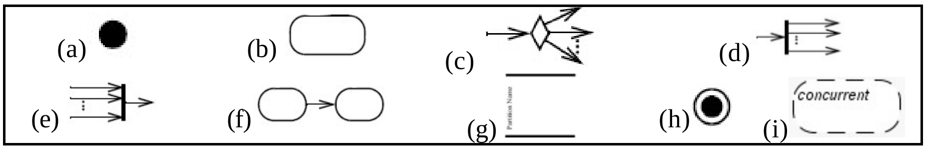

Figure 2: UML Activity Diagrams - From right to left (a) InitialNode – a signal indicating a start point in a process; (b) Action – refers to an atomic activity; (c) DecisionNode; (d) ForkNode or AND-Split; (e) JoinNode; (f) ControlFlow (g) ActivityPartition or Swimlane; (h) ActivityFinal; (i) Expansion region meaning that the actions defined inside of that will be repeated in an specific number determined by parameter.

2.1 Document Approval Pattern

A document approval process constitutes a set of agreements whereas each agreement is performed by one organizational role. The approval process is completed when all organizational roles have finished their revisions or one of these roles does not agree with the document content.

As illustrated in Figure 3, an organizational role "reviewer" performs a document review either resulting in an approval or disapproval. The document review activity is performed multiple times in parallel or in sequence according to the number of organizational roles specified or until a disapproval occurs. Generally, the number of organizational roles is connected to the level of centralization with respect to decision making. The authority to make decisions in organizations can be less or more centralized. In the first case, individuals at the top of the organizational chart have the highest authority to make decisions. The authority of other individuals is delegated top-down according to their position in the organizational chart. The more centralized the authority is the bigger is the number of org. roles involved following the organizational scalar chain. The latter specifies who is subordinated to whom within an organization (Davis, 1996).

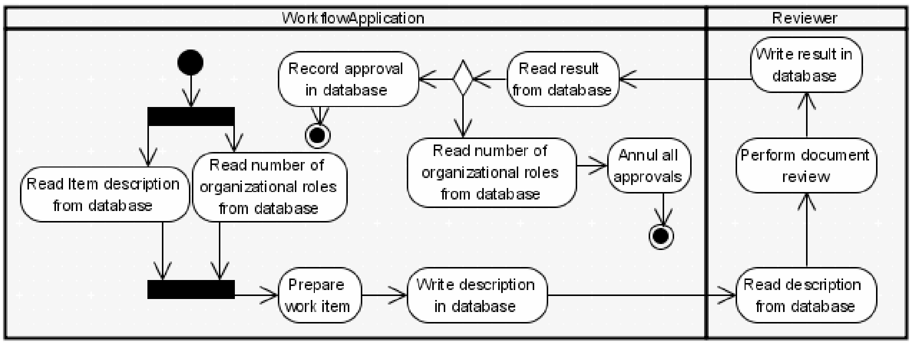

Figure 3: Approval pattern

2.2 Unidirectional Performative Message Pattern

A sender uses unidirectional performative messages to request the execution of an activity from a receiver. Figure 4 shows the pattern: an activity execution request results in a work item being assigned to a receiver (i.e., a specific workflow participant responsible for activity execution). After that, the process may continue execution without waiting for a response. Note that the unidirectional performative message does not require a response.

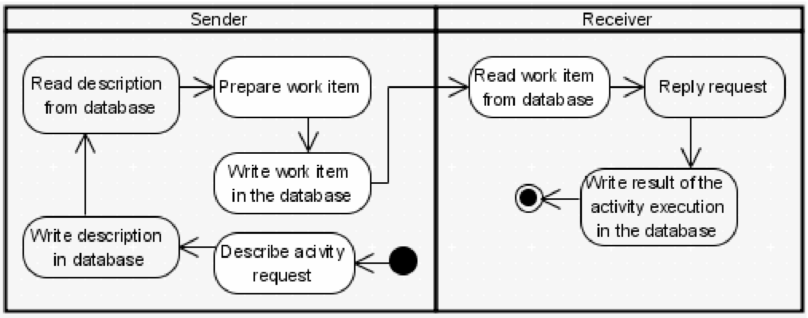

Figure 4: Unidirectional performative message pattern

3 Evidencing the Existence of Workflow Patterns Through Pattern Mining

With the goal to search for the existence of the workflow patterns in real applications we mined 190 workflows. These workflows have been modeled with the Oracle Builder tool and have stemmed from 12 different organizations related to different application domains. Note that the mining was based on the analyses of these workflows and not on corresponding instances or logs generated by the execution of them. Table one characterizes the workflows which were subject of the mining.

Table 1: Core characteristics of the workflows analyzed during the mining

Size of the company	Kind of decision-making	Examples workflows we analyzed	Number of workflows analyzed
1 small	Decentralized	Management of internal activities	17
1 large	Decentralized	TQM and management of activities	11
6 large	Centralized	TQM; control of software access; document management	133
4 large	We had no access to information about these companies	Help Desk, User feedback; document approval	29

We have obtained the following results from these mining activities:

a) evidence with high probability that the workflow patterns presented in (Thom, 2006) exist in real workflows;

b) evidence that the set of patterns is both necessary and sufficient to model all 190 workflows analyzed; and

c) identification of a set of rules that do not only define specific workflow patterns but also show how they are combined with existent control flow patterns (e.g., sequence, XOR-Split).

3.1 Method of Pattern Mining Used

For each workflow pattern we calculated its support value (*S*). In the context of this paper, *S* represents the number of occurrences of each pattern (*P*) in a set of 190 workflows. For those processes comprising more than one occurrence of the same pattern just one was considered. The following formula was considered to calculate the support:

$$S = \frac{F(P)}{T_T}$$

Where:

F(P) = frequency of a specific pattern in the total set of workflows

T_T = total number of workflows

Initially, we identified and annotated workflow patterns in all workflows we analyzed. Figure 5 shows an example of this pattern identification procedure.

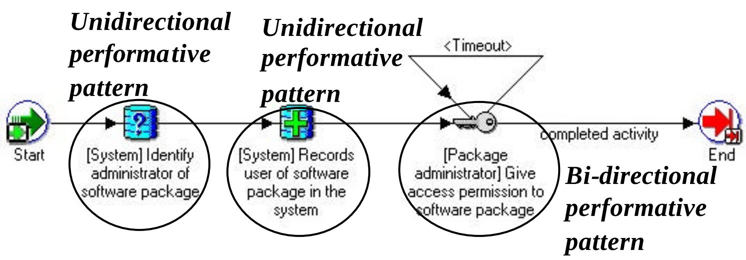

Figure 5: Real process that contain the workflow patterns

Afterwards, for all workflows we counted the number of occurrences of each pattern. The obtained result then was divided by the total number of analyzed processes (i.e. 190 in our case). Accordingly, the *(P)* for this calculation corresponds to a specific pattern while T_T means the set of workflows.

3.2 Frequency of the Workflow Patterns in Real Workflows

Patterns based on recurrent functions (e.g., *unidirectional* and *bi-directional performative patterns*, *decision pattern*, *notification pattern* and *informative pattern*) are not dependent on specific application domains or organizational structure aspects. This fact mainly explains why they were identified with high-probability in practically all workflows analyzed. The same applies to the *document approval pattern*. This can be explained by the high degree of centralization on decision-making existing in the organizational units for which we analyzed their workflows. This high centralization implies the use of approval activities (cf. Section 2.1). Besides that, several workflows belong to applications related to approval contexts. By contrast, most of the workflows analyzed do not comprise *question-answering* activities.

As most of the workflows analyzed neither comprise *logistic* nor *financial* activities, the corresponding patterns are practically not present. The financial pattern, was identified in activities comprising some internal monetary attribute that is used by the next activity in the process. Indeed, none of the workflows include the *logistic pattern*. Figure 6 graphically illustrates the frequency of each pattern in the set of workflows analyzed (i.e. the *support* value).

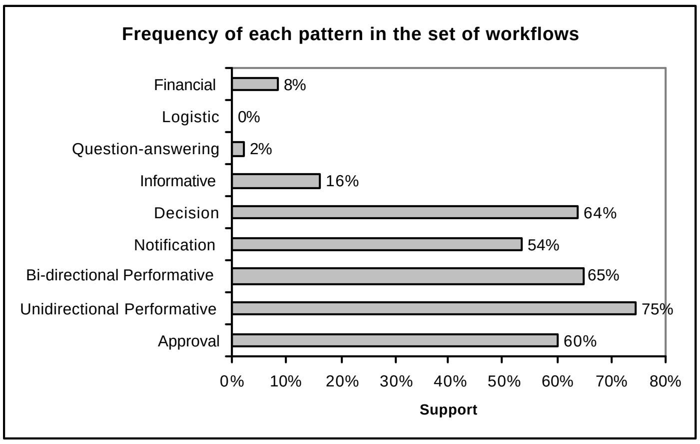

Figure 6: Frequency of workflow patterns in real workflows

We also investigated the frequency of selected patterns to specific characteristics of the activities where they were identified. First, we analyzed the purpose of each activity and identified the most related pattern. Afterwards, we annotated and counted the kind of the activity (i.e., automatic or manual). In our last step we identified the subsequently control flow connected to the activity. Table 2 summarizes the results of this investigation. It shows, per example that 97 of the analyzed approval processes (i.e., more than 85% of the total number of processes) can be defined in terms of a composition of a *bidirectional performative pattern* in a *manual activity* followed by an *Exclusive Choice (XOR-Split)* control flow pattern. Based on such information we defined a set of rules that connect selected patterns with specific control flows (see Thom, 2006b).

Table 2: Specific characteristics of workflow patterns

	Purpose of the activity	Kind of activity	Subsequently control flow	*S* value
Unidirectional	Task execution request with no mandatory response	Manual	Sequence	Workflows = 142 Support(*S*) = 99%
Bi-directional	Task execution request with mandatory response	Manual /automatic	XOR-Split	Workflows = 123 *S* = 100%
Decision	Specific selection instead of an evaluation or approval	Automatic	Bi-directional + XOR-Split	Workflows = 132 *S* = 92%
Notification	Activity result, user advice or remember	Manual /automatic	Unidirectional + Sequence	Workflows = 102 *S* = 100%
Informative	Information request (e.g., user filling out a form)	Manual	Unidirectional + Sequence	Workflows = 31 *S* =100%
Approval	Approval activity	manual	Bi-directional + XOR-Split	Workflows = 97 *S* = 85%

6 Conclusions

This paper has introduced workflow patterns. Each of them is based on a frequent process activity (e.g., a task execution request, notification activity, approval).

The main goal of the presented mining procedure was to measure the frequency with which each workflow pattern occurs within the set of analyzed workflows. This analysis was accomplished in order to verify whether respective workflow fragments may be considered as patterns with high probability for reuse.

The main difficulty of this mining effort was the non-availability of tools for automatically identifying the workflow patterns within the set of analyzed workflows. Such a tool could reduce the mining time and the human effort significantly.

What really surprised us was the fact that all analyzed workflows can be defined as a composition of the investigated patterns. That is, the set of workflow patterns is necessary and sufficient to design all 190 real workflows that were subject of the mining effort. In each process, a specific workflow pattern may appear zero or more times combined with other patterns.

This fact can be considered as a very important one, which indicates new research directions for future work. For example, to which degree may this pattern set be helpful when being integrated into a workflow design tool? One could think of an intelligent software module which relies on both a workflow patterns repository and the set of patterns combining rules (e.g., *approval pattern* = [*bi-directional performative pattern* + *manual activity*] → *XOR-Split*). Such rules could be applied in the process of translating legacy software applications (e.g., those ones written in COBOL) into workflow based applications with activities that are written in modern programming languages.

References

Aalst, W.M.P. van der; Hee, K. van: *Workflow Management*: models, methods, and systems. London: The MIT Press (2004).

Bradshaw, D.; Kennedy, M.; West, C: *Oracle BPEL Process Manager. Developer's Guide* (2005).

Berbstein, P. A.; Hadzilacos, V.; Goodman, N.: *Concurrencly Control and Recovery in Database Systems*. Reading: Addison-Wesley (1987) 370p.

Davis, M. R.; Weckler, D. A: *A Practical Guide to Organization Design*. Boston: Crisp Publications (1996).

Flores, F.: *Computer Systems and the Design of Organizational Interaction* (1998).

Keller, G.; Nüttgens, M.; Scheer; A.-W.: *Semantische Prozessmodellierung auf der Grundlage "Ereignisgesteuerter Prozessketten (EPK)"*. Heft 89, Institut für Wirtschaftsinformatik, Saarbrücken, Germany (1992).

Malone, T. W.; Crownston, K.; Herman, G. A.: *Organizing Business Knowledge* (2004).

Medina-Mora, R.: *The action workflow approach to workflow management technology* (1992).

Mendling, J; Aalst, W. M. P. van der: Towards EPC Semantics based on State and Context. In: *Proceedings of the 5th EPC Workshop EPK 2006*, CEUR Workshop Proceedings (2006) pp.25-48.

Muehlen, M. zur: *Workflow-based process controlling*: foundations, design, and application of workflow-driven process information systems. Berlin: Logos Verlag (2002) 299 p.

Russell, N.: *Workflow Resource Patterns*. Brisbane: Queensland University of Technology (2004).

Russell, N.; Hofstede, A. H. M Ter; Edmond, D.: Workflow Data Patterns. In: *Proceedings of the 24th ER* (2005) pp. 353-368.

Russell, N.; Aalst, W.M.P. Van Der; Hofstede, A. Ter.: Workflow Exception Patterns. In: *Proceedings of 18th CAiSE*, 18 (2006) p.288-302.

Thom, L. H.; Iochpe, C.; Amaral, V. L. do; Viero, D. M. de.: Toward block activity patterns for reuse in workflow design. In: *Workflow Handbook of WfMC* (2006) pp. 249-260.

Thom, L. H.: *A Pattern –based Approach for Business Process Modeling*. PPGC: Porto Alegre (2006). Available at: http://www.biblioteca.ufrgs.br/bibliotecadigital/. (Thesis).

Thom, L. H., Lau, J. M., Iochpe, C., Mendling, J. Extending Business Process Modeling Tools With Workflow Pattern Reuse. Prof. Int'l Conf. on Enterprise Information Systems, Funchal, Funchal (2007) (to appear).

Thomas, O.; Scheer, A.-W.: Tool Support for the Collaborative Design of Reference Models - A Business Engineering Perspective. In: *39th HICSS-39 2006, CD-ROM / Abstracts Proceedings* (2006).

Workflow Management Coalition (WfMC): *Process Definition Interface*: XML Process Definition Language. Doc. Number: WFMC-TC-1025 (2005).