# Towards an Intelligent Workflow Designer based on the Reuse of Workflow Patterns

[1]Cirano Iochpe, Carolina Chiao[1],
Guillermo Hess[1], Gleison Nascimento[1]
Federal University of Rio Grande do Sul
Institute of Informatics
Av. Bento Gonçalves, 9500
91502-970, Porto Alegre, Brazil
+55 (51) 33166820
{ciochpe, cchiao, hess,gsnascimento}@inf.ufrgs.br

Lucinéia Thom[2],
Ulm University
Institute of Databases and Information Systems
Oberer Eselsberg
Geb. O27, Raum 515, 89069
+49 (073) 15024137
lucineia.thom@uni.ulm.de

Manfred Reichert[3]
Faculty of Eletrical Engineering mathematics and Computer Science
University of Twente
7500 AE Enschede
The Netherlands
+31 (53) 489 3705
m.u.reichert@cs.utwente.nl

## ABSTRACT
In order to perform process-aware information systems we need sophisticated methods and concepts for designing and modeling processes. Recently, research on workflow patterns has emerged in order to increase the reuse of recurring workflow structures. However, current workflow modeling tools do not provide functionalities that enable users to define, query, and reuse workflow patterns properly. In this paper we gather a suite for both process modeling and normalization based on workflow patterns reuse. This suite must be used in the extension of some workflow design tool (e.g., Intalio, Event Process Chain – EPC). The suite comprises components for the designing of processes from both legacy systems and user process.

## Keywords
Workflow Patterns reuse, Business Process, Legacy Systems, Process Design, Ontology.

## 1. INTRODUCTION
For several reasons companies are developing a growing interest in improving the efficiency and quality of their internal business processes and in optimizing their interactions with customers and business partners. During the last years we have seen an increasing adoption of business process management tools by enterprises as well as emerging standards for business process specification in order to meet these goals. Respective technologies (e.g., workflow management systems) enable the definition, execution, and monitoring of the operational processes of an enterprise. In connection with Web service technology, in addition, the benefits of business process management from within a single enterprise can be transferred to cross-organizational business processes as well.

Business Processes and respective workflow models frequently include a variety of fragments (*or recurrent business functions*) which can be understood as self-contained activity blocks with a specific and well-defined semantics [1], [2], [3]. As an example, let's consider the evaluation process for price adjustment as depicted in Figure 1. This process includes activities with the following partial order: (a) verify if it is a shopping order or not; (b) evaluate request of price adjustment; (c) notify managers about conclusion of evaluation; (d) notify managers about automatic approval. Altogether this process comprises fragments that presents generic semantics which can be described as patterns such as decision (activity *a*), notification (activities *c* and *d*), and task execution request (activity *b*). In this paper, we are dealing with the question of how the modeling of processes that include recurrent business functions like notification in Figure 1 can be supported appropriately by a tool. Moreover, how the extraction of business process from legacy systems could be accomplished when supported by such patterns.

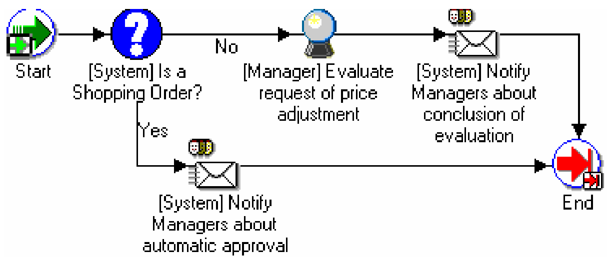


**Figure 1. Evaluation process for price adjustment**

So far, several workflow patterns have been suggested for representing control flow [5], resources [6], data [7] interaction [8] and exception handling [9]. Yet, these pattern sets have in common that they are relevant for the implementation of a workflow system and the definition of process modeling languages, but they provide only a partial answer to the question of what business functions a modeler has to consider repeatedly in various process models. Usually, such process *fragments* [10], [11], [12], [13], [8] are re-designed for practically every workflow application. Such procedure can be considered as inefficient, and thus undesirable from a maintenance perspective. While there is some research reported on how metadata can be organized to manage large-scale modeling project (see [14]), we are not aware of any work evidencing the existence of recurrent patterns in real workflow applications as well as their necessity and completeness for the business and workflow process modeling. Besides that, contemporary workflow modeling tools do not provide functionalities that enable users to define, query, and reuse such patterns in a proper way.

Related to these problems we proposed a set of seven workflow patterns in an early work [1]. Each pattern represents a recurrent business function (such as the ones showed in Figure 1) frequently found in business processes. In this paper we introduce a suite for normalizing and modeling of business processes based on the reuse of workflow patterns. By normalization we mean the definition of a standard description form to which the business processes are translated, i.e., a canonical format for describing workflows.

This suite (the so called "Intelligent Workflow Designer") must be used in the extension of some workflow design tool (e.g., Intalio, Event Process Chain – EPC), and is intended to provide a number of functionalities, such as: (1) the extraction of business processes from legacy systems and their normalization, correctness checking in a formal notation and translation into a standard notation; (2) support for process design, by suggesting to the modeler patterns to be combined to the one he/she modeled and; (3) construction of a knowledge base for storage and retrieval of workflow patterns.

Against this background, the outline of this paper is organized as follows: Section 2 gives an overview of the workflow patterns that we identified in prior researches. In particular, we discuss the approval, unidirectional performative and the notification pattern as three examples. Section 3 gathers a suite for the extension of some modeling tool that aims to support the reuse of these patterns. We describe each component of the suite which considers process modeling from legacy systems or from user processes (e.g., design from scratch). 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 7 patterns from an extensive study based on the literature. Examples of patterns are document approval, question-answering, unidirectional and bi-directional performative, information, notification and decision patterns. Details on these patterns as well as a classification of them are reported in [2] and [3].

It is out of the scope of this paper to detail the semantics of all these patterns. It is important to note that through the mining of 190 real workflow processes we measured the occurrence frequency of each of the workflow patterns in the set of workflow processes analyzed. In general words, the main results of the mining can be summarized as follows:

- There is a high probability that the workflow patterns exist in real workflow processes, i.e., 60% of the analyzed workflow processes include organization-based patterns; 8% include some domain application–based patterns; and 75% include patterns related to such business functions;
- The set of patterns appears to be both necessary and sufficient to model all the 190 real workflow processes analyzed. From this, one can conclude that the detected patterns could be very suitable for defining both business processes and workflows related to different application domains. In addition, with few patterns it is possible to design a large variety of workflows which we believe can reduce complexity and design effort (see [4]).
- A set of rules that not only define specific workflow patterns but also show how they are combined with existent control flow patterns (e.g., sequence, XOR-Split). The rules are described in [4].

We illustrate the approval, unidirectional performative and the notification pattern as examples.

### 2.1 Examples of Workflow Patterns

A block activity is suitable to represent each pattern according to [15]. 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.

Since the patterns representation may require input/output parameters and the block activity concept does not support parameters, the transaction perspective of serialization theory was applied to overcome this limitation [16]. 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 the three example patterns as an UML Activity Diagram (using the UML 2.0 notation). The Visual Paradigm for the UML Community Edition based on UML 2.0 was used as an editing tool to design the patterns. Figures from 3-5 must be read according to the legend in Figure 2.

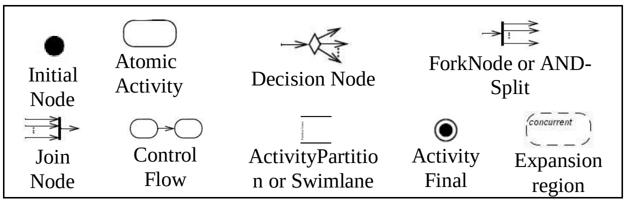


**Figure 2. UML Activity Diagrams**

#### *2.1.1 Document Approval Pattern*

**Context:** A doccument must be evaluated by one or more organizational role.

**Problem:** How to model a human decision-making in the workflow process?

**Forces:**

- The number of times that the decision-making activity is repeated may vary depending on the level of centralization of authority (less or more) as well as the direct supervision of work existent in the organizational unit(s) where the process is executed.

- The decision-making activity must be performed by a human.
- The decision-making activity must have more than one kind of response (e.g. approval and repproval).

**Solution**: To include in the workflow, a human activity that characterizes a point of decision-making on the sub-product in question (e.g. a document requiring approval). Figure 3 shows the process construct.

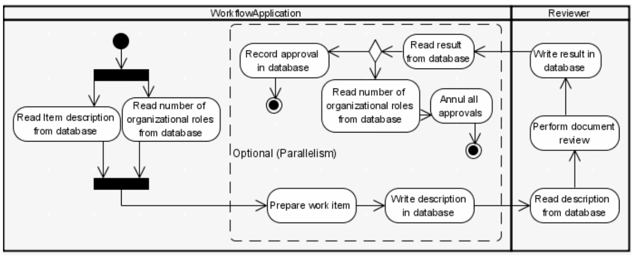


**Figure 3. Approval pattern**

In Fig. 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 (in high positions of the organization) with respect to decision-making.

### *2.1.2 Unidirectional Performative Pattern*

**Context:** In a workflow, there is a moment that the process must request the execution of an activity to the system or to an organizational role involved in the process.

**Problem:** While modeling the workflow, how to send a request without waiting for the result of the activity execution?

**Forces:**

- The receiver's response is not required.
- The process must keep its execution without waiting for the activity to be completed.
- The request can be done to the system or a human.

**Solution:** To include in the workflow model the sequence of activities (see Fig. 4) representing the unidirectional performative message. The sequence of activities comprises the generation of a work item in the receiver worklist. However, the workflow does not wait for a receiver response to continue execution.

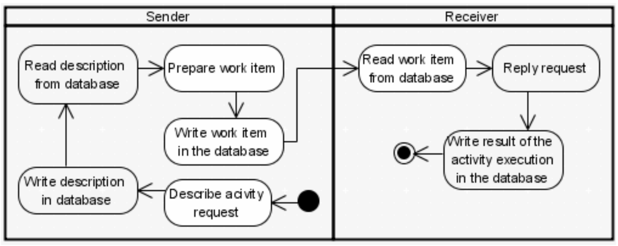


**Figure 4. Unidirectional performative message pattern**

A sender uses unidirectional performative messages to request the execution of an activity from a receiver. As shown in Fig. 4, 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.

### *2. 1.3 Notification Pattern*

**Context:** During the process execution, some specific events have more relevancy such that the process must inform some organizational roles about them.

**Problem:** While modeling the workflow, how can we keep the involved roles informed about some process instance events?

**Forces:**

- The notification must be sent by an electronic way.
- The process does not have to wait for a reading response to keep its execution.
- The notification must contain the status of a process activity which the monitoring is desired.

**Solution:** To include in the workflow the structure concerning the notifying and do not wait for response. Figure 3 shows how this structure works. There's the sending of the notification and its receiving. These activities must inform the involved roles in the process about news inherent to the respective workflow, such as, for example, the approval (or repproval) of a document, or a task that has achieved its timeout.

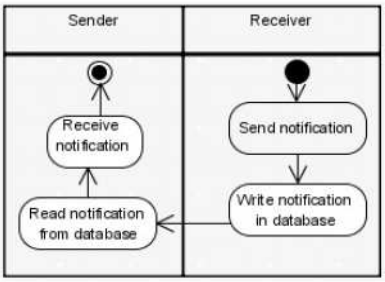


**Figure 5. Notification pattern**

## 2.2 Core Characteristics of the Workflow Patterns

We also investigated the frequency of the selected patterns to specific characteristics of the activities where they were identified. First, we analyzed the purpose (content) of each activity and identified the most related pattern (e.g. approval activity → approval pattern). Afterwards, we annotated and counted the type of the activity (i.e., automatic or manual). In our last step we identified the subsequent control flow connected to the activity. Table 1 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 [2]).

**Table 1. Specific characteristics of workflow patterns**

| | Kind of activity | Subsequent control flow | Frequency (%) of subsequent control flow |
|---|---|---|---|
| Unidirectiona l | Manual | Sequence | Workflows = 142 i.e., 99% |
| Bi-directional | Manual /automatic | XOR-Split | Workflows = 123 i.e., 100% |
| Decision | Automatic | XOR-Split | Workflows = 132 i.e., 92% |
| Notification | Manual /automatic | Sequence | Workflows = 102 i.e., 100% |
| Informative | Manual | Sequence | Workflows = 31 i.e., 100% |
| Approval | manual | XOR-Split | Workflows = 97 i.e., 85% |

## 3. THE INTELLIGENT WORKFLOW DESIGNER

This Section describes an approach towards the development of a suite for both process modeling and normalization based on the reuse of workflow patterns such as the ones presented in Section 2.1). We intend to use this suite in the extension of some workflow design tool (e.g., Intalio, Event Process Chain – EPC). We believe that our patterns can improve the correctness of workflow design as well as reduce design efforts. In [4] we present the first inside in this direction.

The core functionalities of the Intelligent Workflow Designer are:

1. Extraction of normalized business processes from legacy systems: Comprises the extraction of business rules from the analysis of source code (e.g. COBOL, clipper, access, visual basic, C++) of legacy systems and subsequent generation of business processes in high-level notation (such as the Business Process Modeling Notation - BPMN). The process is then, validated by matching it with existent workflow patterns (such as the ones Section 2.1 presents) stored in a knowledge database. The challenge here is to identify all embody patterns comprised by the process. As a result the process is translated into one or more patterns (such as the ones proposed in [4]). Such procedure must benefit the translation of the processes to some execution language (e.g., BPEL4WS). Furthermore, with the scope of business process extraction, a model checking is performed, in order to test the correctness (accuracy) of the process.
2. Support to process design: a user process is received by the intelligent workflow designer as an input. The process is then, matched with patterns stored in the knowledge database in order to identify the partial order of patterns it comprises. Having this information, the intelligent designer will suggest the most suitable patterns that are feasible to be used subsequently by the already designed process. In addition, it will inform how often each pattern combination was used in earlier modeling.
3. Construction of a knowledge database of workflow patterns: The workflow patterns repository (ontology) will store not only the patterns but also the frequency with each pattern is most feasible to be combined with other patterns (e.g., control flow patterns). Through the mining of new processes we believe that such frequencies can be improved in order to increase their precision. Thus, in design time the accuracy, concerning the frequency associated with each suggestion of combined patterns be correct may increase. Figure 6 illustrates the suite.

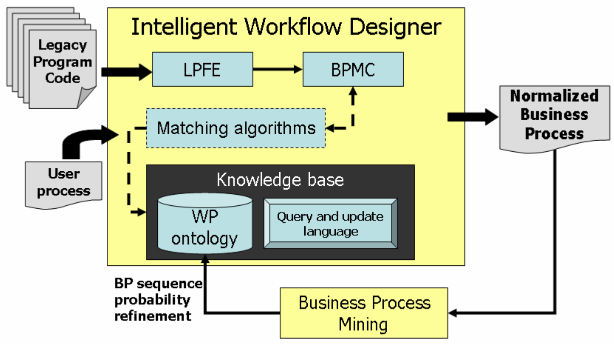


**Figure 6. Intelligent Workflow Designer Suite**

Core components of the Intelligent Designer Workflow are:

- **Legacy Program Flow Extractor (LPFE)**: component responsible by the extraction of business process rules from the source code of legacy systems. Moreover, generation of corresponding process in high-level language (such as Business Process Modeling Notation - BPMN).
- **Business Process Model Checking (BPMC)**: this component verifies how complete and correct the extracted process is. First the process is translated to some formal language (e.g., Pi-calculus). In case it is correct, the process is matched with the knowledge base so that the patterns comprised by the process can be identified.
- **Matching Algorithms**: algorithms responsible by the identification (*matching*) of the workflow patterns stored in the ontology. The selected patterns are those comprised by either the user process or the processes generated by the LPFE and BPMC components.
- **Knowledge Base**: is the database where the workflow patterns are stored. It is composed by an ontology which describes the patterns. It also comprises a query and update language (mechanism). This mechanism is useful to identify the most suitable patterns (concerning earlier use frequency in modeling) to be used subsequently of a given pattern (based on earlier probabilistic mining results). In addition, the update mechanism must be used to change the probabilistic results of each sequence of patterns based on the process mining results.
- **Business Process Mining**: External tool to the Intelligent Workflow Designer which receives a set of normalized workflow patterns as input. The output is then, used to update the knowledge database, in special the use frequency of each pattern.

## 4. CONCLUSIONS

While workflow patterns were defined for several aspects related to process execution, the aspect of recurrent business functions is only partially addressed by existing work. In prior work, we identified a set of seven workflow patterns that appear necessary and sufficient to model an extensive set of workflows from practice. In other work we investigated in how far process modeling tools can be tailored to provide a direct support for pattern reuse. Currently, we are also working in the documentation of the patterns with Pi-calculus. Moreover, we are analyzing the sequences of workflow patterns in real workflow processes in order to study the probabilities of the possible sequences.

In this paper we investigate in how far process modeling tools can be tailored to provide a direct support for patterns reuse. We consider that business process can be either designed from scratch or extracted from legacy systems existent in organizations. Our contribution is a suite to the analysis and properties verification in workflow specification (e.g., correctness, completeness, deadlock, processes equivalence, livelock, model checking). This suite must be used in the extension or developing of some workflow tool in order to makes feasible the modeling of business process from patterns reuse.

The main advantages of this approach can be summarized as follows: (a) the completeness and necessity of the workflow patterns for the workflow process design had already been evidenced in prior work; (b) the suite is tool-independent and can be adapted for any business process modeling tool; (c) the business process model checking can be considered as a very important component which can help in the verification of how complete and correct is the process that is being designed, specially through the matching with patterns stored in the knowledge base.

As future work we not only intend to improve the architecture presented in this paper but also to implement it through the development or extension of some workflow design tool. We strongly believe that the use of the workflow patterns introduced in this paper must not only reduce design effort but also improve the correctness of it [3]. Following this trend, we expect to perform additional experiments concerning process design with and without the workflow patterns. These experiments are quite important to demonstrate how process design (from legacy system or from scratch) can be improved.

## ACKNOWLEDGEMENTS

The authors would like to acknowledge the Coordination for the Improvement of Graduated students (CAPES), the Institute of Databases and Information Systems of the University of Ulm (Ulm, Germany) and the Informatics Institute of Federal University of Rio Grande do Sul (Porto Alegre, Brazil).

## REFERENCES

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

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

[3] Thom, L. H.; Iochpe, C.; Reichert, M. Workflow Patterns for Business Process Modeling. In: Workshop on Business Process Modeling, Development, and Support (BPMDS'07). Trondheim, Norway. (2007a).

[4] Thom, L. H; Lau, J.M.; Iochpe, C.; Mendling, J. Extending Business Process Modeling Tools with Workflow Pattern Reuse In: Proceedings of 9th International Conference on Enterprise Information Systems (ICEIS 2007), Funchal, Madeira, Portugal (2007b).

[5] Aalst, W.M.P. van der. Workflow Patterns. In Distributed and Parallel Databases, 14(3), pages 5-51, (2002).

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

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

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

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

[10] Flores, F.; et al. Computer Systems and the Design of Organizational Interaction. (1998).

[11] Medina-Mora, R. The action workflow approach to workflow management technology. (1992).

[12] Malone, T. W.; Crownston, K.; Herman, G. A, Organizing Business Knowledge. (2004).

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

[14] 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).

[15] Workflow Management Coalition. Terminology & Glossary. Bruxelas, (1999), 65p. Available at: <http://www.wfmc.org>. Visited on Nov. 2005.

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