APPLYING BLOCK ACTIVITY PATTERNS IN WORKFLOW MODELING

Lucinéia Heloisa Thom, Cirano Iochpe
Institute of Informatics, Federal University of Rio Grande do Sul, Av. Bento Gonçalves 9500, Porto Alegre, Brazil

Keywords: Business process types, block activity, workflow patterns, business/workflow process modeling.

Abstract: This paper discusses a set of business (sub-)process types represented as "workflow block activity patterns". We describe all patterns in a common language (UML 2.0) and through some study cases we tried to find out whether they are frequently reused during business/workflow process modeling. The results showed that the patterns are frequently identified in both workflow elements and workflow applications. We believe they can be reused to improve the quality as well as the performance of the design phase in a workflow project.

1 INTRODUCTION

In the last years, research on workflow patterns has increased mainly because of the reuse advantages that patterns can establish (Thom, 2005), (Hohpe, 2004). The most expressive approaches are in the field of control/data flow patterns (Aalst, 2003), (Russell, 2004) as well as resource and application – oriented patterns (Russell, 2004b). However, a lot less research can be found relating workflow design to a set of recurrent business process "pieces" that must be atomically executed by the workflow process. Although one can precisely characterize the semantics of such business process "pieces" (Medina, 1992), (Malone, 2004), (Muehlen, 2002) and they have to be recurrently re-designed in practically every workflow modeling process, there is no known research relating these business process structures to workflow patterns.

1.1 Approach

An activity set is a self-contained set of activities and transitions (WfMC, 2005). Transitions in the set should refer only to activities in the same set and there should be no transitions into or out of the set. Activity sets can be modeled as block activities. The block execution starts at the first activity in the set and executes next activities by following the partial order established upon them by the transitions until it reaches an exit activity. Workflow execution then returns to the next activity following the block.

The approach we undertake in this paper applies the block activity concept in order to represent a set of business process types found in the literature which we call then "workflow block activity patterns". According to (Gamma, 1995) a pattern is the abstraction from a concrete form which keeps recurring in specific non-arbitrary contexts.

The block activity concept is suitable for representing the selected business process "pieces" because it can encapsulate their well-defined semantics as well as represents their atomic characteristic. It means that all activities defined inside the block activity pattern must execute before workflow continues execution.

Since our patterns representations may require input/output parameters and the block activity concept does not support parameters, we apply the transaction perspective of the serializability theory to overcome this limitation (Bernstein, 1987). An input parameter is represented as a database read operation of a "one-time-only" readable information. Similarly, an output parameter is represented in the block as a database write operation of a "one-time-only" writeable information.

As part of our investigation, after integrating different pattern classifications found in the literature, the so-called workflow block activity patterns were described in a common language. The UML 2.0 (OMG, 2005) was chosen for this purpose. In order to verify whether these workflow patterns are modeled as often as their counterparts in business processes, we conducted some case studies by analyzing both a set of workflow definition

Heloisa Thom L. and Iochpe C. (2006).
APPLYING BLOCK ACTIVITY PATTERNS IN WORKFLOW MODELING.
In *Proceedings of the Eighth International Conference on Enterprise Information Systems - ISAS*, pages 457-460
DOI: 10.5220/0002448004570460

languages and a set of workflow processes.

1.2 Related Work

Wil van der Aalst proposed 21 workflow patterns for the description of business process behavior (Aalst, 2003). More recently, Aalst proposed a set of workflow data patterns (Russell, 2004) and a set of resource workflow patterns (Russell, 2004b). Our approach differs from Aalst' approach because each of our pattern have content and context based on specific business functionality (e.g., task execution request and informative communication)

SAP created a cross-application tool called SAP Business Workflow. The tool makes feasible the integration of business tasks between applications including a workflow wizard with workflow templates (Andrews, 2003).

The Massachusetts Institute of Technology (MIT) started in 1991 the Process Handbook development, an online knowledge base including entries for over 5000 business activities (Malone, 2004). We consider our patterns more application independent then SAP and MIT patterns.

The remainder of the paper is structured as follows. Section 2 brings an overview about different business process types. Section 3 describes some of these process types through UML 2.0. Aiming to investigate whether they are implemented in workflow components of different workflow tools as well as different applications a matching exercise was performed. The results of it are summarized in Section 5. Finally, Section 6 concludes the paper and brings future directions.

2 BUSINESS (SUB-)PROCESS TYPES

The participants of a process communicate by exchanging *messages*. According to (Hope, 2004) a message exchange includes two roles: a sender or producer (sends a message) and a receiver or consumer (receives a message). Furthermore, a message comprises two parts, a header and a body. The header contains meta-information about the message (e.g., who send it and where it is going). The body is formed by: (a) a data; (b) an activity execution/information request or; (c) a notification.

In (Muehlen, 2002) messages are classified in *unidirectional* and *bi-directional*. Unidirectional messages are used either by a sender to *request* the execution of an activity from a receiver, or by a receiver to *notify* a sender. Bi-directional messages either form a *request/respond* pair, where a sender asks a receiver to perform an activity and the receiver answers the sender, or they form a *solicit/respond* pair, where a receiver asks the sender for information which is supplied subsequently.

Communication processes are characterized by the exchange of messages between at least two participants of the process. In the literature we found four mainly different kinds of communication processes (performative unidirectional/bi-directional, informative and notification) (Geurts, 2004).

A *performative communication process* refers to an activity execution request being either unidirectional or bi-directional. By the other hand, an *informative communication process* comprises an information (e.g., data) required by some process participant, i.e., aims at the sharing of existing knowledge (Muehlen, 2002), (Medina, 1992). A *notification process*, in our approach differs from an informative communication process because it comprises only process execution status.

When a process includes physical tasks it is called *material process* (Medina, 1992). The manufacturing, storage and transport of physical objects are examples of material processes. However, when the process is characterized by tasks such as the buying and selling of goods it is called *logistic process* (Muehlen, 2002). By the other hand, when a monetary value is exchanged between two parties it is called *financial process* (Muehlen, 2002). Finally, the process can includes a *decision* making, i.e, a cognitive process of selecting a course of activities from among multiple alternatives (OMG, 2005).

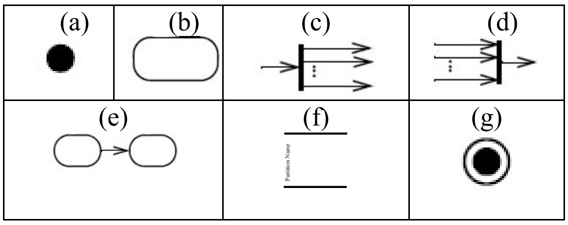

Figure 1: From right to left (a) InitialNode; (b) Action – refers to an atomic activity; (c) ForkNode or AND-Split; (d) JoinNode; (e) ControlFlow (f) ActivityPartition or Swimlane; (g) ActivityFinalNode.

3 BLOCK ACTIVITY PATTERNS

Within the context of a business as well as workflow process there are a variety of different business process "parts" which can be understood as self-contained activity blocks with a specific and well defined semantic. It is worth observing that the same "part" can be repeated within the same process. At execution time different copies of a same "part" may be receiving either the same or different parameter values.

Due to space limitation, this section presents only 2 patterns in UML 2.0 notation. We rely on the block activity concept of the WFMC to model structures which should be executed atomically. Figures 2 and 3 must be read according to the legend presented in Figure 1. We used the Visual Paradigm for UML Community Edition based on UML 2.0 as an editor tool to design the patterns.

3.1 Unidirectional Performative Communication Pattern

This pattern represents a unidirectional performative communication process. As shows in Figure 2, firstly there is an activity execution request. Based on the activity description, a work item is assigned to a receiver. After that, the process can continue execution without waiting for response. The write as well as read activities would be modeled as parameters if allowed by a block activity.

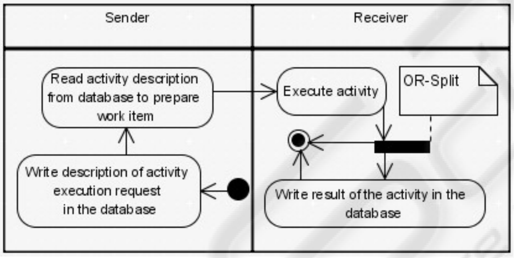
Sender
Receiver
Read activity description from database to prepare work item
Execute activity
OR-Split
Write description of activity execution request in the database
Write result of the activity in the database

Figure 2: Unidirectional performative communication pattern.

3.2 Notification Pattern

The notification pattern (illustrated in Figure 3) is based on the notification process type. It comprises a notification activity that can either inform the end of an activity execution or post news inherent to the workflow application. In the last case, the sender usually sends a notification. In our approach we are treating the notification activity as a self-contained activity because we consider that a notification activity status can, eventually, be sent after previously being requested to do so.

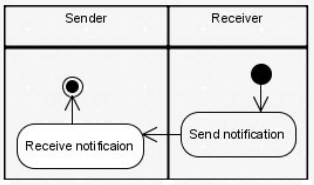

Figure 3: Notification pattern.

4 CASE STUDY

Aiming at identifying the patterns introduced in Section 3 in real processes we performed a "matching exercise". Firstly we searched for individual as well as combinations of patterns in the workflow modeling language supported by the Oracle Workflow Cartridge (Oracle Guide, 2001). After that, we tried to find the same patterns within a set of real workflow processes that were modeled as well as implemented with that tool. In a third step we analyzed the set of JBoss (Jboss, 2005) modeling elements. Due to the lack of space, in the following we present only an example concerning the process for the evaluation of new products that are supposed to be in the market. The process (see Figure 4) starts with a notification to a process participant who is responsible for the launching evaluation. After that the system notifies the Financial department whether the product was accepted or not.

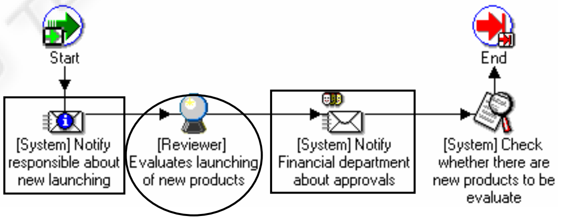
Start
End
[System] Notify responsible about new launching
[Reviewer] Evaluates launching of new products
[System] Notify Financial department about approvals
[System] Check whether there are new products to be evaluate

Figure 4: Launching of new products.

In this process we identified the notification patter (represented as a square) and the bidirectional performative communication.

5 CONCLUSIONS

This paper presented an overview of the main business process types found in the literature represented as "workflow block activity patterns". Through some study cases they were identified both in workflow components from different tools and in workflow applications. The main results of the "matching exercise" were: (a) the patterns could be identified in most of the analyzed workflow elements. Accordingly, we perceived that they are probably present in workflow applications developed with bases on such workflow elements; (b) the patterns were also identified in specific "pieces" of a workflow application which makes feasible their reuse in similar new applications.

By applying the WfMC "activity block" concept in the patterns definition we provide the atomicity property, meaning that whole activities inherent to a specific patter are completely executed from beginning to end before the flow (outside the activity block) can continue. Additionally, the serializability theory was also suitable to cover the parameter expression limitation of the block activity.

Each of our patterns present a well-defined and usefulness semantic referring to some specific "part" of the business (sub-)process. We also highlight that our patterns are conceptual level -oriented patterns, thus they are suitable to be used during the workflow design phase. With the patterns being proposed we aim to improve both the quality and the performance of the modelling phase in a workflow project mainly because of the reuse advantages of pattern approaches.

As future work we intend to continue investigating both workflow elements of different tools and workflow applications of different domains. The advantage of that approach is twofold: firstly it showed to be an interesting way to evaluate the workflow tools; secondly, it can leads to the discovering of new patterns.

ACKNOWLEDGEMENTS

The authors acknowledge the German Academic Exchange Service – DAAD, the Coordination for the Improvement of Graduated Students – CAPES (Brazil), the Institute for Parallel and Distributed Systems – IPVS of University of Stuttgart (Stuttgart, Germany) and the Informatics Institute of Federal University of Rio Grande do Sul – UFRGS (Porto Alegre, Brazil).

REFERENCES

Aalst, W.M.P. van. et al. 2003. Workflow Patterns. *In Distributed and Parallel Databases*, 14(3), pages 5-51.

Andrews, T. et al. 2003. *Business Execution Language For Web Services*, (Version 1.1). Bernstein, P. A.; Hadzilacos, V.; Goodman, N. 1987. *Concurrency Control and Recovery in Database Systems*. Reading: Addison-Wesley, 370 p. : il

Eriksson H.-E., Penker, M. 2001. *Business Modeling with UML*. John Wiley & Sons, Inc., 459p.

Gamma, E. et al. 1995. *Design Patterns*. Addison-Wesley.

Geurts, G.; Geelhoed, A. 2004. *Business process decomposition and service identification using communication patterns*.

Hohpe, G.; Woolf, B. 2004. *Enterprise integration patterns*: Designing, Building, and Deploying Messaging Solutions. Boston: Addison-Wesley.

Jboss jBP 3.0. 2005. *Workflow and BPM made pratical*. User Guide.

Malone, T. W.; Crownston, K.; Herman, G. A. 2004. *Organizing Business Knowledge*: The MIT Process Handbook. ISBN 0-262-13429-2.

Medina-Mora, R. et. Al. 1992. *The action workflow approach to workflow management technology*. Available in : www.acm.org.

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

Object Management Group (OMG). 2005. *Unified Modeling Language: Superstructure*. V.2.0.

Oracle Workflow Guide. 2001. Realise 2.6.2. v. 1. 1056p.

Russell, N., Hofstede, A. H. M ter, Edmond, D. 2004. Workflow Data Patterns. In: *Informatik 2004 - Informatik verbindet (Band 1)*, Lecture Notes in Informatics (LNI). Proceedings... Ulm. v. p-50.

Russell, N. et. al. 2004b. *Workflow Resource Patterns. Technical report*, FIT-TR-2004-01, Queensland University of Technology, Brisbane.

Thom, L., Iochpe, C., Mitschang, B. 2005. Improving the Workflow Project Quality Via Business Process Patterns Based on Organizational Structure Aspects. In: *1st GI Workshop XML4BPM - XML for Business Process Management* at BTW, Karlsruhe Germany.

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