ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В УПРАВЛЕНИИ ПРОИЗВОДСТВЕННЫМИ СИСТЕМАМИ

ОБРАБОТКА ИСКЛЮЧИТЕЛЬНЫХ СИТУАЦИЙ В СИСТЕМАХ УПРАВЛЕНИЯ ПОТОКАМИ ДАННЫХ ПРЕДПРИЯТИЯ

Ю.О. Гончарук
(г. Томск, Томский политехнический университет)
E-mail: divia777@mail.ru

EXCEPTION HANDLING IN WORKFLOW MANAGEMENT SYSTEMS

Y.O. Goncharuk
(Tomsk, Tomsk Polytechnik University)

Annotation. This paper considers different aspects of exceptional situations definition and handling during workflow system implementation process.

Introduction. Definition of Workflow

The term “workflow” usually is used as a synonym for “business process”. Usually specialists recommend using the terminology defined by the Workflow Management Coalition (WFMC). This is an organization whose main aim is to develop standards, definitions and standard interfaces for workflow management systems modules.

Workflow can be defined as automation of a business process in whole or in part, during which documents, information or tasks are passed from one participant to another for action, according to predefined set of procedural rules, to achieve a common goal [1].

A workflow consists of an orchestrated and repeatable pattern of business activity enabled by the systematic organization of resources into processes that transform materials, provide services, or process information. It can be depicted as a sequence of operations, declared as work of a person or group, an organization of staff, or one or more simple or complex mechanisms [2].

Exception management in workflow systems

An important feature of workflow systems is the ability to represent exceptions that alter the normal flow of processes. Among exceptions, a class which is gaining recognition and importance is that of expected exceptions, i. e., of those anomalous situations that are known in advance to the workflow designer.

When an exception is unexpected, the exception handler typically resorts to halting the process and invoking a human intervention. Instead, when exceptions are expected, the exception handler can rely on the semantics of the workflow application in order to handle the exception, typically by means of some form of reactive processing. For instance, in a car rental workflow, an accident to a rented car causes an exception to the regular rental process.

The accident, although expected, is an unlikely event; once it has occurred, however, a variety of activities become needed, including, e.g., giving assistance to the renters, scheduling the car’s repair, and rescheduling the future rentals for the affected car. All such activities constitute the (planned) reactions to raising the exception. Workflows describe the “normal behavior” of a process, while expected exceptions model the “occasional behavior”.

Expected exceptions are unpredictable, and therefore cannot be conveniently represented in the process in the form of special tasks and connections among tasks. They are not frequent, but once they occur they may require special treatment, which may lead to the execution of a completely different process.

They are asynchronous (hence, initiated at an arbitrary stage of the process) and highly influenced by external factors. Their execution may cause the backtracking of previous steps in the process or even sudden termination [3].

There is growing interest and need for languages and systems to integrate workflow specifications with expected exceptions; commercial workflow systems typically support only a selected number of them, without enough generality.

Some kind of comprehensive approach to the management of expected exceptions was created; it was defined a new language for expressing expected exceptions, and then describe the features of a system for integrating the exception handler with the workflow manager. For brevity, it is used the term “exception” to denote asynchronous expected exceptions.

The exception-handling mechanism must be able to capture exceptional events and to react to them. Each reaction must first assess the state of the process and then, in a few cases, adopt the corrective action; in many cases events correspond to false alarms and do not need to be followed by a corrective action. This model has a strong similarity to the trigger management strategy used in active databases.

In fact, since most workflow systems execute on top of commercial databases, it is quite natural to use active database functionality to manage exceptions. Active rules are characterized by the following components, each with an immediate correspondence to exceptions:

1. The event part defines the symptoms of an exception, e.g., database modifications or signals coming from other components of the workflow, that trigger the rule, i. e., put it in the set of rules to be considered by the rule management system.
2. The condition is a boolean predicate that checks that the symptoms really identify an exception to be managed; it can also be used to select, among several exception management alternatives, the most adequate to deal with the current workflow state.
3. The action describes the updates and procedures that must be invoked to respond to the exception occurrence [4].

Each rule is executed in a new transactional context, different from the one in which the exceptional event was generated; in terms of active databases, rules have a detached execution mode.

In most cases, rules do not need immediate service and should interfere as little as possible with regular workflow processing. Thus, it was engineered an exception handler in which triggered rules are batched and considered at given periods of time. Very few events are classified as “real time” and cause an immediate invocation of the rule management system.

The effort invested by the Workflow Management Coalition in the interchangeability of process definitions has led to the definition of the XPDL language, a commonly acknowledged XML format for process definition. While XPDL effectively enables the cross-product portability of process definitions, the language has not been designed to also capture undesired behaviors that may arise during process execution, i. e. exceptions [5].

To alleviate process designers from the inherent complexity of enriching existing process definitions with exception handling constructs by hand, it was developed a suitable compiler, which enables the designer to model the plain process as usual, to specify the exceptions to be handled in form of Chimera-Exception triggers, and to automatically compile the trigger definition into the enriched process definition.

Nonetheless, exceptions – especially those that are predictable at process definition time – do have semantics that are not negligible. Due to the very tight interactions between the basic exception management constructs (the sub-processes) and the workflow engine of the host WfMS in order to suitably capture events and to enact actions, the development of the library of basic subprocesses typically requires intimate knowledge of the host WfMS which is very important step during workflow system implementation.

Список литературы

1. Grefen, P. Database Support for Workflow Management: The Wide Project / P. Grefen, B. Pernici // Kluwer International Series in Engineering and Computer Science, № 4, 1999. – P. 31–50.
2. Nastansky L., Hilpert W. The GroupFlow System: A Scalable Approach to Workflow Management between Cooperation and Automation. – Berlin: Springer, 1994. – p. 264.

3. Fisher, L. The Workflow Handbook / L. Fisher // Workflow Management Coalition (WfMC), 2004. – 641 p.
4. Van der Aalst, W. Workflow Management Models, Methods, and Systems / W. Van der Aalst, K. Van Hee // MIT Press, 2004. – 853 p.
5. Weske, M. Workflow Management Models, Methods, and Systems / M. Weske // MIT Press, 2004

ОЦЕНКА ЭФФЕКТИВНОСТИ СИСТЕМЫ ПРОДАЖ В РОЗНИЧНОМ БИЗНЕСЕ

Т.Р. Рахимов, Я. Русснер
(Томск, Томский политехнический университет
Кемниц, Германия, Университет Технологий)

EFFICIENCE EVALUTION OF SELLING SYSTEMS IN RETAIL BUSNES

T.R. Rakhimov, J. Russner
(Tomsk, Tomsk Polytechnic University
Kemnitz, Germany, University of Technology)

Abstract: efficiency evaluation in retail business is important indicator of competitiveness. This article presents efficiency evaluation of a selling system of a Supermarket, explains the obtained results and suggests ways to improve it.

Efficiency evaluation is one of the key activities of a manager in retail business. High numbers of companies in retail business create strong levels of competition which puts emphasis on price and cost competition. It results in greater focus on creating more efficient operations systems. Efficiency evaluation is part of the control function and provides important input for planning function in company's management cycle.

The purpose of this article is to evaluate efficiency of an existing selling system in retail business, using the example of Lama Supermarket in Trade Center "Smile City". To achieve this goal the following tasks had to be accomplished: 1) Development of a system of efficiency indicators for evaluation. 2) Obtaining factual, relevant data on the selling system of Lama Supermarket through monitoring 3) Efficiency evaluation based on obtained monitoring results.

Efficiency of operating systems should not be confused with their productivity. Efficiency is a narrower concept that pertains to getting the most out of a fixed set of resources; productivity is a broader concept that pertains to effective use of overall resources [1].

System of efficiency indicators for any particular system is developed in accordance with the following basic principles. It should specific, measurable, adequate, transparent, representative, stable, cost-efficientetc.

In accordance with the purpose we decided to conduct a study of efficiency evaluation applied to selling systems in retail business. For that purpose as an object under consideration Lama Supermarket was selected. This store is situated in the city of Tomsk, in trade center "Smile City".

The system of efficiency was formed to include the following indicators [1].

Design Capacity–is the maximum output rate or service capacity an operation, process, or facility is designed for.

Effective Capacity – is a design capacity minus certain allowances such as personal time, maintenance, and scrap.

Design capacity is the maximum rate of output achieved under ideal conditions. Effective capacityis usually less than design capacity owing to realities of changing product mix, the need forperiodic maintenance of equipment, lunch breaks, coffee breaks, problems in scheduling andbalancing operations, and similar circumstances.