

Business process exception handling using workflow methods

Dovilė VOJEVODINA (VGTU)

e-mail: d.vojvodina@it.lt

1. Introduction

E-business refers to the usage of Internet technologies, to improve and transform key enterprise processes [1]. That means, e-business is used to connect suppliers, business partners and customers to the enterprise critical business systems, so that they could access useful information they need. The need of e-business comes from the customer oriented politics, need to supply all business members with current and correct information, technologies continuing to evolve to support this type of environment. People prefer to gain information they need using Internet technologies because they can access several manufacturers or service suppliers without leaving their homes. They can collect and compare information in purpose to choose the best item they need. Evolution of money into e-money also gives a reason to transform daily business processes into e-business processes.

So every enterprise, which seeks to lead in their sphere, transforms their daily tasks into e-business processes, so that they later could be integrated into the core business applications.

2. E-business

In case to understand the nature of exceptions and propose it's solution, it is necessary to become acquainted with the e-business nature also. The difference between e-business and traditional business is Internet characteristics and communications using it. It is possible:

- to communicate with many clients at the same time, because e-business systems are autonomous;
- support doesn't require many people - there is dual relation between consumer and enterprise, which makes easy to track good's traffic (which is a great help solving exception cases).

This paper is dedicated to the inter-business applications, which are used for communication between enterprise and other organizations or customers. That would be business-to-business (B2B) applications, which are for business partners; and business-to-consumer (B2C) applications, which are for customers [12].

B2C applications are for consumers only and based around the Internet. The consumers are outside the organization so they are using organization services, which can be goods or information.

B2C applications are frequently modeled using workflow because these tasks are frequent and accomplished according to set of rules and input data. Workflow makes it possible to process raised process instance quickly, using correct data and present user with the correct and effective solution.

Business process workflow always has an opportunity to extend. There is always a possibility to change the chain of tasks and activities or add a sub-process into existing process. But that applies only to the future business process instances. As for the old instances, they must be left as they are, or they may be altered using other workflow processes, which alters old instance to meet new instance requirements and supply with necessary data.

B2B applications transform inter-organizational relationships and are dedicated for partnership. They can be used to exchange information between businesses and for collaboration, sharing information and knowledge between businesses for mutual benefit [12].

In B2B case, communication is held between enterprise and business partners. Both of them can have their individual e-business applications, which are based on workflow methods. So, both of them need to have exception handling applications, which should be able to track and support as e-business application exception, as application integration with other enterprise intra/inter applications exceptions.

Still, there is possibility for exception to occur in communication spot, between enterprise and business partner applications. In this case, both exceptions handling applications must evaluate situation and act accordingly. Both, enterprise and business partner must be explained about the situation and results.

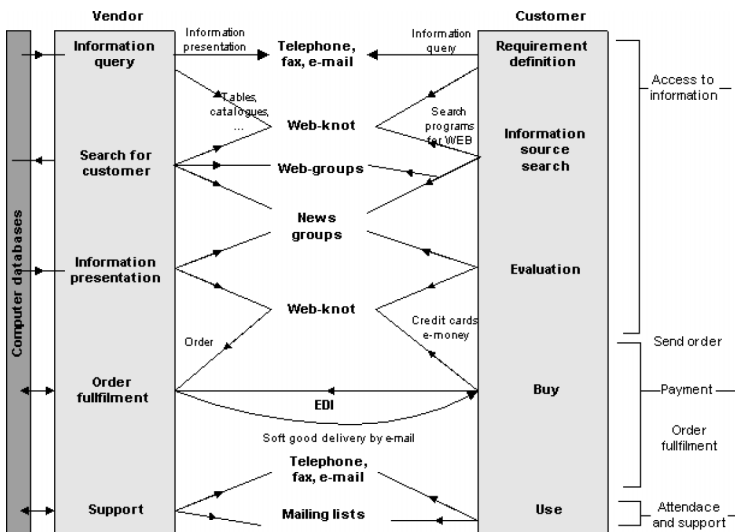


Fig. 1. E-business and business processes.

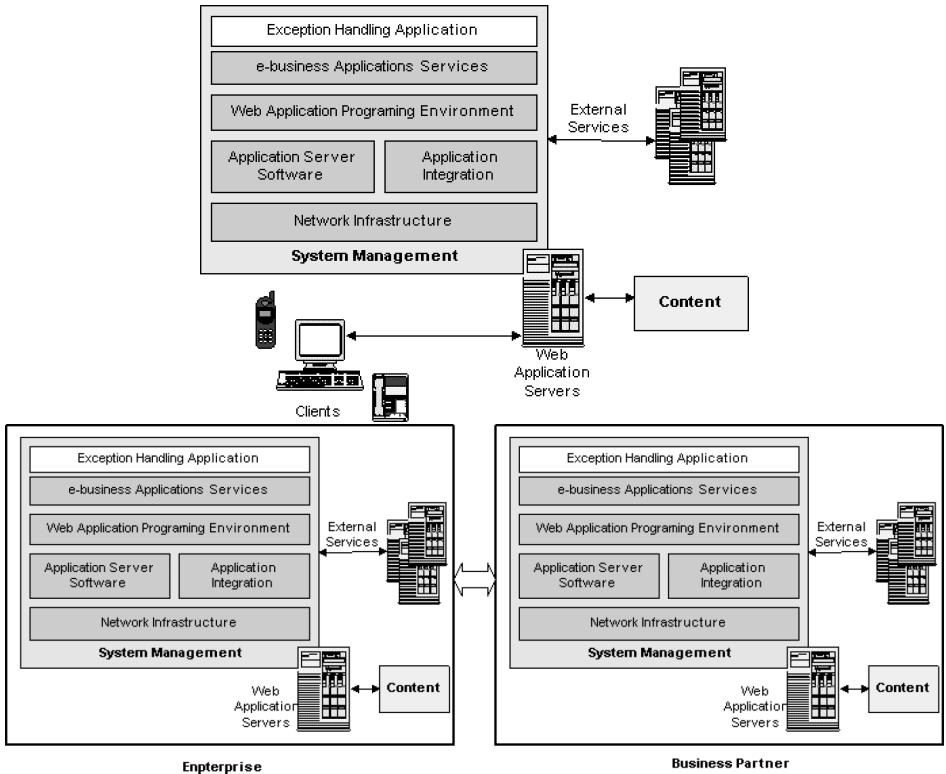


Fig. 2. Business-to-Consumer Application Framework for e-business architecture.

3. Exception classes and sources

In case, to construct exceptions handling mechanism, we need to classify exceptions and their sources. The nature of exceptions can be very various. In the first place, it is necessary to draw an ideal business process model. If we use a workflow for that purpose, it must be a dedicated graph with all its possible branches, which are already designed. It is possible to predict most possible exceptions or to determine the spots there they might occur, from this graph. But that would be only expected and known exceptions. Still there is a possibility for unexpected and unknown exceptions. Though expected exceptions in a way can't be called exceptions, rather business process cases. Expected and known exceptions are always modeled during business process modeling time and become part of all business process.

Some of described above exceptions can be ascribed to the workflow exception or application exception class. Workflow exceptions cover system exceptions and user-defined exceptions. All expected and know exceptions are usually workflow exceptions, because they occur in workflow management system. Unknown and unexpected exceptions are more application level exceptions, because here are raised process instances, which can raise different and unpredicted deviations from ideal business process model. This level

exceptions are closely tied to every task and are dependent on application level semantics, but still, exception which occur here and was unpredicted can raise exception in the infrastructure level. Latter exceptions ascribe to server, DBMS, Web or similar exceptions, which are managed by other than business process exception handling mechanism, but by server or DBMS error handling mechanisms. It would be more accurate to call infrastructure level exceptions simply hardware errors [6].

All of exceptions can occur synchronously with the flow of the process or asynchronously. That means, either they occur together with the process, by executing it, or some task or instance raise an exception, which at the beginning affects unknown parts of the business process and only later raises synchronous to some process exception. Of course it can happen vise-versa.

From all of this exception classification it's clear that the only exceptions that can be handled and doesn't impact the work model are known exceptions, otherwise called as useful exceptions [5]. All other exceptions must be handled at the time of their occurrence or later and not always solved.

4. Handling

An instance, which is unable to complete, identifies the process not specified in the workflow model, that is, an exception [7]. That means, to handle exception, one has to know where it occurred, why it occurred and how to solve it. In other words it is exception detection, diagnosis and resolution.

Detection phase always starts from the model of correct behavior, both for entire system and for each individual component [8]. This model must be drawn such, which achieves business process goal and is considered ideal process model.

In case to find an exception in this ideal model one has to have a taxonomy of elementary process elements, which can be annotated with failure modes. In [10] this taxonomy consists of exceptions related to goals and assumptions, exceptions related to activities and exceptions related to resources. To find an exact place of an occurred exception, this taxonomy must be broken up into smaller specified units.

To specify the possible exception mode, the system can be instrumented with additional sentinel components, which would look for appropriate patterns in the behavior of base components [7]. We also can use similar sentinels to collect information during all the process execution time, in case to collect information about it's work, so that it would be later used for inspection of occurred exception. Such method would help to step over the detection phase and go straight to the diagnosis phase.

Diagnosis phase is used to determine which methods or tools should be used to solve unacceptable situation. In this phase exception specifications can be processed, which are generally made in the detection phase. The results are used to make the exception handler specification. For the systems efficiency exception handler should be generated by the autonomous process or system, which could have an exception and exception handlers' knowledge base. In this way, information and solutions are collected all the time and reused if possible in the future. This would lessen exceptions handling time and cost, which are critical for every business process model.

In generally, exception detection and diagnosis needs a lot of precise information, so this information can be also stored using ontology. It can be used to describe the ideal business process model in the terms of the ontology, so that it clear communication between system developers and domain experts [11]. In the ontology-based case management there are used three phases: abstraction, retrieval and adaptation [6][9]. Each phase uses different specialized ontology to represent exception case, or to retrieve similar cases or to adapt the existing cases. Use of ontology clarify and frame business process terminology and logic, so that the exception handling can be achieved more precise.

Exception handlers are usually modeled as autonomous processes and are always enacted in the place of exception occurrence, not appearance. These processes can implement three different types of actions: masking, propagation and recording [6]. These actions can be used separately or together. The exception process can ignore the exception, modify the process according to changed data and record the situation for the future reuse. Also exception process can retry corrupted task, execute appropriate workflow recovery operation or modifications.

The basic goal of any exception handling situation is to solve an exception in such a way, that it would be possible to return to the ideal process' exception spot and continue process without it's change or effect to him.

Exception handling problems leads to the assumption that any business model must be able to learn, in other words to know how to collect information, where and why. So exception handling system must be able to: react properly to the unpredictable situations, explain its performance understandably to the user and be customized in a flexible, fast way with justifiable effort.

In e-business applications, sometimes it is necessary to exchange the process description, exception case description information between the companies. While modeling e-business applications it is necessary to use a common data format, which would be appropriate for both sides, enterprise and consumer/business partner. XML(Extensible Markup Language) is a good solution because it has a strong data description capabilities, it enhances delivery and representation. Also XML data can be easily transformed into other data format if it is necessary. Once both sides agree to use XML as a standard for the data exchange, the communication can be made much easier. Also XML can be used not only for the data exchange between enterprise and business partners or consumers, but also inside enterprise system management, in purpose to exchange data in the same format between intra/inter applications.

XML can be as source format for content, as a document exchange format and as a document delivery technology. There are a lot of advantages for that, such as flexible data interchange using metadata, integration data from other sources, enhanced search and filtering capabilities, automatic document validation, managing smaller, reusable fragments of documents, rich presentation formats.

5. Conclusions

E-business is growing and getting more and more popular because of it's benefits to the both sides: enterprise and consumer. Workflow methods and technologies are used to

make in hole or part such applications. Because of it's nature, workflow is clear understandable to business people and programmers, so better performance of the application can be reached. If workflow is used to describe and execute business processes, it makes sense to use workflow for it's exception handling. Again, exception handling must be understandable as to consumer, as to application, so workflow methods makes it clear for both. Finally, if communication has a form understandable to both sides, it lessens the chances for exceptions to occur.

References

- [1] I. Naick, M. Berkhoff, D. Verdugo Bosnich, IBM Redbooks, *Business-to-Business Integration Using MQSeries and MQSI Paterns for E-Business Series*, ITSO, IBM Corp (2002).
- [2] K. Rutledge, *The Business Case for E-Business*, IBM Press (2000).
- [3] *Microsoft Press* (1999).
- [4] *Workflow Management Coalition, Terminology & Glossary*, WPMC-TC-1011, Issue 3.0. Winchester Hampshire, United Kingdom (1999).
- [5] W.S. Sadiq, M.E. Orłowska, *On Capturing Exceptions in Workflow Process Models*, The University of Queensland, Australia, Department of Computer Science and Electrical Engineering.
- [6] Zongwei Luo, A. Sheth, K. Kochut, J. Miller, Exception handling in workflow systems, *Applied Intelligence*, **13**(2), 125–147 (2000).
- [7] S.W. Sadiq, O. Marjanovic, M.E. Orłowska, Managing change and time in dynamic workflow processes, *International Journal of Cooperative Information Systems*, World Scientific Publishing Company.
- [8] C. Dellarocas, Toward exception handling infrastructures in component-based software, in: *Proceedings of the International Workshop on Component-Based Software Engineering, 20th International Conference of Software Engineering (ICSE)*, Kyoto, Japan, April 25–26 (1998).
- [9] A. Maedche, B. Motik, L. Stojanovic, R. Studer, R. Volz, Ontologies for enterprise knowledge management, *IEEE Computer Society*, 1094-7167/03 (2003).
- [10] C. Dellarocas, M. Klein, A knowlwdge-based approach for handling exceptions in business process, *Information Technology and Management*, **1**(3), 155–169 (2000).
- [11] M. Augustaitytė, *Investigation of Ontology Usage Problems in the Software Development Process*, MCs thesis, Department of Information Systems, Vilnius Gediminas Technical University, Vilnius (2001).
- [12] L. Ennsler L, P. Leo, T. Meszaros, E. Valade, *IBM Redbooks, The XML Files: Using XML for Business-to-Business and Business-to-Consumer Applications*, ITSO, IBM Corp. (2000).

Verslo procesų nukrypimų valdymas darbų sekų metodu

D. Vojevodina

Verslo procesų kompiuterizavimas sudaro organizacijoms sąlygas apibrėžti savo veiklos tikslus, siekti geresnio darbo ir galutinio produkto rezultato, įgyti klientų palankumą. Tam įgyvendinti dažnai pasirenkami darbų sekų metodai. Darbų sekos leidžia nesunkiai apibrėžti verslo procesus taip, kad jie būtų aiškūs ir suprantami tiek programuotojui, tiek verslo atstovams. Gaila, bet darbų sekos nepalaiko dinaminės verslo procesų prigimties. Kitaip tariant, negali savarankiškai spręsti jokių darbų sekų pakeitimų bei žinomų ir nežinomų situacijų. Šis straipsnis yra skirtas e-verslo darbų sekų nukrypimų tvarkymo problemoms ir sprendimams aptarti.