

## Chapter 4.13

# Multiple-Step Backtracking of Exception Handling in Autonomous Business Process Management

**Mingzhong Wang**

*University of Melbourne, Australia*

**Jinjun Chen**

*Swinburne University of Technology, Australia*

**Kotagiri Ramamohanarao**

*University of Melbourne, Australia*

**Amy Unruh**

*University of Melbourne, Australia*

### ABSTRACT

This chapter proposes a multiple-step backtracking mechanism to maintain a tradeoff between replanning and rigid backtracking for exception handling and recovery, thus enabling business process management (BPM) systems to operate robustly even in complex and dynamic environments. The concept of BDI (belief, desire and intention) agent is applied to model and construct the BPM system to inherit its advantages of adaptability and flexibility. Then,

the flexible backtracking approach is introduced by utilizing the beneficial features of event-driven and means-end reasoning of BDI agents. Finally, we incorporate open nested transaction model to encapsulate plan execution and backtracking to gain the system level support of concurrency control and automatic recovery. With the ability of reasoning about task characteristics, our approach enables the system to find and commence a suitable plan prior to or in parallel with a compensation process when a failure occurs. This kind of computing allows us to achieve business goals efficiently in the presence of exceptions and failures.

DOI: 10.4018/978-1-60566-669-3.ch010

## INTRODUCTION

A critical challenge in building practical business process management (BPM) systems is to allow users to maintain the system robustness and reliability with respect to the correct execution even in the presence of abnormalities. As the operating environment becomes increasingly complex, dynamic and error-prone, it is extremely challenging for designers and programmers to find out all possible combinations of exceptions as well as designing corresponding handling methods. Therefore, a flexible, systematic and autonomic approach for exception handling is essential for the success of applying complex BPM systems into wider fields of application.

Multi-agent systems have been extensively studied as a powerful high-level decomposition and abstraction tool in analyzing, designing, and implementing complex software systems (Jennings, 2001). Many researchers and practitioners have noticed the fundamental relationship between agents and workflow systems (Ehrler, Fleurke, Purvis, & Savarimuthu, 2006) and proposed various approaches to build dynamic and adaptive workflow systems with the concept of agents, and vice versa.

To manage the complexity arisen from the dynamic and complex running environment, we propose to apply the reactive BDI (Belief, Desire and Intention) (Rao & Georgeff, 1995) agent system to model and construct BPM systems, thus benefiting from its sound features of event-driven and means-end reasoning for the purpose of robustness, flexibility and adaptability.

Providing a higher level of abstraction, agents help to simplify modeling and construction of flexible BPM systems operating in open and distributed environments. To deal with exceptions, most multi-agent systems apply the mechanism of backtracking to recover and retry the failed execution. However, they back track in a rigid step-by-step manner until one alternative ex-

ecution path is found. Moreover, they follow a defensive route of recovery-then-try pattern. Due to the dynamic and nondeterministic features of complex environments, rigid recovery on the reverse chronological order of execution history will in many instances become meaningless and inappropriate. Some attempt to consider starting a fresh plan after the exception. However, this approach is too computationally expensive and rarely practical to get deployed because it requires a complete knowledge about the running environment and needs to consider all previous actions. As a result, developers are forced to consider low-level details of disturbances, failure, or uncontrolled interactions between workflow actors for the requirement of robustness and reliability.

To address this issue, we propose to extend existing rigid backtracking strategy to support and enable execution backtracking in a multiple-step fashion (reverse chronological order with certain steps skipped). Instead of step-by-step backtracking through the execution tree, the system can “jump” back to an arbitrary history node and try another eligible execution path. Our approach maintains a tradeoff between replanning and traditional backtracking strategies.

To provide automatic and system-level support for concurrency control and exception handling, open nested transaction (Weikum & Schek, 1992) is integrated into our multiple-step backtracking model. Compared with other approaches which apply transaction models (Gray & Reuter, 1993) into workflow (Georgakopoulos, Hornick, & Manola, 1996) or multi-agent systems (Ramamohanarao, Bailey, & Busetta, 2001; Nagi, 2001), our method unites the beneficial features of event-driven and means-end reasoning from BDI agent systems and utilizes a flexible backtracking approach to allow the execution “jumping” back several levels at once to continue its execution towards the goal in case that backtracking to one level in the execution tree does not solve the problem.

16 more pages are available in the full version of this document, which may be purchased using the "Add to Cart" button on the publisher's webpage:

[www.igi-global.com/chapter/multiple-step-backtracking-exception-handling/48599](http://www.igi-global.com/chapter/multiple-step-backtracking-exception-handling/48599)

## Related Content

---

### Optimizing Both the User Requirements and the Load Balancing in the Volunteer Computing System by using Markov Chain Model

Abdeldjalil Ledmi, Hakim Bendjenna and Hemam Sofiane Mounine (2018). *International Journal of Enterprise Information Systems* (pp. 35-62).

[www.irma-international.org/article/optimizing-both-the-user-requirements-and-the-load-balancing-in-the-volunteer-computing-system-by-using-markov-chain-model/198429](http://www.irma-international.org/article/optimizing-both-the-user-requirements-and-the-load-balancing-in-the-volunteer-computing-system-by-using-markov-chain-model/198429)

### A Novel Finger-Vein Recognition Based on Quality Assessment and Multi-Scale Histogram of Oriented Gradients Feature

Junying Zeng, Yao Chen, Yikui Zhai, Junying Gan, Wulin Feng and Fan Wang (2019). *International Journal of Enterprise Information Systems* (pp. 100-115).

[www.irma-international.org/article/a-novel-finger-vein-recognition-based-on-quality-assessment-and-multi-scale-histogram-of-oriented-gradients-feature/220401](http://www.irma-international.org/article/a-novel-finger-vein-recognition-based-on-quality-assessment-and-multi-scale-histogram-of-oriented-gradients-feature/220401)

### Message-Based Approach to Master Data Synchronization among Autonomous Information Systems

Dongjin Yu (2012). *Enterprise Information Systems and Advancing Business Solutions: Emerging Models* (pp. 194-209).

[www.irma-international.org/chapter/message-based-approach-master-data/66576](http://www.irma-international.org/chapter/message-based-approach-master-data/66576)

### E-Commerce Security

Steven Furnell (2006). *Enterprise Information Systems Assurance and System Security: Managerial and Technical Issues* (pp. 131-149).

[www.irma-international.org/chapter/commerce-security/18385](http://www.irma-international.org/chapter/commerce-security/18385)

### The Validation Process for Tier-III Variables and Fitness of the Model

(2013). *Managing Enterprise Information Technology Acquisitions: Assessing Organizational Preparedness* (pp. 238-254).

[www.irma-international.org/chapter/validation-process-tier-iii-variables/76980](http://www.irma-international.org/chapter/validation-process-tier-iii-variables/76980)