DENNIS WOLF

INVESTIGATING EVENT SUBSCRIPTION MECHANISMS IN BPMN

INVESTIGATING EVENT SUBSCRIPTION MECHANISMS IN BPMN

DENNIS WOLF

Hasso Plattner Institut
Digital Engineering • Universität Potsdam

< Any Subtitle? >

August 2017 – version 1

ABSTRACT

Business Processes have become an essential tool in organizing, documenting and executing company workflows while Event Processing can be used as a powerful tool to increase their flexibility especially in destributed scenarios. The publish-subscribe paradigm is commonly used when communicating with complex event processing platforms, nevertheless prominent process modelling notations do not specify how to handle event subscription.

At the example of BPMN 2.0, the first part of this work illustrates the need for a flexible usage of event subscription in process models and derives new requirements for process modelling notations. An assessment of the coverage of these requirements in BPMN 2.0 is presented and shortcomings are pointed out.

Based on the identified requirements, this work presents a new concept for handling event subscription in business process management solutions, predominantly built on the notion of event buffers. The concept includes an extension to the BPMN meta model, specifies the semantics and API of a new event buffering module and describes the changes necessary to process engine behaviour.

For evaluation purposes, the concept has been implemented as a reusable Camunda Process Engine Plugin that interacts with the academic Complex Event Processing Platform UNICORN.

ZUSAMMENFASSUNG

Kurze Zusammenfassung des Inhaltes in deutscher Sprache…

CONTENTS

LIST OF FIGURES

LIST OF TABLES

LISTINGS

ACRONYMS

1 INTRODUCTION

2 BACKGROUND

3 PROBLEM STATEMENT

This section will further define the problem and derive formal requirements to event subscription mechanisms

3.1 MOTIVATING EXAMPLES

- one example for independent. The subscription does not depend on a prior precess result, the subscription can be done even before process instantiation

- one example for a process that uses an intermediate event that depends (subscription-wise) on the result of a previous step in the process.

3.2 EVENT SUBSCRIPTION SCENARIOS

What is meant by Event Subscription Scenario and why are we going to work with well-defined scenarios? When can events happen in relation to the process (instance) lifecycle? => timeline When must subscription happen to catch these events? Second dimension: Does the subscription depend on a prior process result? > A matrix of Event Subscription Scenarios.

3.3 REQUIREMENTS DEFINITION

Derive formal requirements, define, make them measurable.

R1: Flexible Event Subscription Time:

R1.1: Explicitness > for each event that is used in a business process, the time of subscription must be clearly defined in relation to the process execution lifecycle > The definition is done in the process model > explicit about the event platform to subscribe to > ? What's the granularity?

R1.2: Flexibility > The subscription time can be chosen at least from the following options: [see scenarios] > Options are limited when the subscription depends on data from previous execution steps

R2: Automatic Subscription Handling

R2.1: Subscription > The subscription to event sources is handled implicitely during process deployment and execution > It is handled according to the modeled subscription time.

R2.2: Unsubscription > The unsubscription from an event source is handled automatically as soon as a subscription becomes unnecessary.

R3: Event Buffering > All events since the subscription time are available to the process. > A single buffer entity can be accessed from different process elements, process instances and processes within the environment > Buffer policies?

4 ASSESSMENT OF THE CURRENT BP MANAGEMENT STACK

What is the goal of this chapter and how does it fit into the structure? Working with the Event Subscription Scenarios.

4.1 EVALUATING BPMN 2.0 AGAINST THE SCENARIOS

(BPMN 2.0 without extensions) For each Scenario: Is it feasible to create a BPMN model that is sound in presence of the given scenario? Define Soundness What is a possible example for this scenario? Provide the BPMN diagram if available. (4 pages)

4.2 IMPLEMENTING EARLY EVENT SUBSCRIPTION USING STANDARD CAMUNDA

Is it possible to implement this using out-of-the-box Camunda? Which aspects cannot be (sufficiently) implemented? How can ... be implemented in Camunda? Show details. Diagrams in appendix. (3 pages)

4.3 DISCUSSION

What are the shortcomings when using out-of-the-box business process solutions to implement Early Event Subscription? What can be implemented without problems? (2 pages)

5 FLEXIBLE EVENT SUBSCRIPTION

Present an abstract framework for flexible event subscription. > Including: Model <> Process Engine <> Buffer <> CEP > How does event subscription currently affect the workflow? > What should a workflow look like that allows early event subscription? > What must be explicitly stated by the user? What should be done automatically in the background? (1 page)

5.1 BPMN EXTENSION

To fulfill requirements R1.1 and R1.2, additional information has to be included in the BPMN model. By default, a BPMN intermediate event does not have information on the time of subscription or the event query. The BPMN specification offers BPMN-X extensions to add custom properties or elements to a model.

To accomodate the required information, the following extension is proposed: > The extension should apply to MessageIntermediateCatchEvent and MessageBoundaryEvent > extend tMessage => tBufferedCEPMessage, so that the messageRef can be reused > OR extension to messageEventDefinition: ExplicitSubscriptionMessageEventDefinition > [subscriptionQuery, subscriptionTime, bufferPolicy]

A buffer shared across multiple instances or events is more complex than a simple single-event-buffer (that one does not require buffer policies). As soon as the requestEvent call can be executed multiple times for the same queryId, we need to specify the following aspects:

Buffer policies: (widely based on [Ref paper Sankalita]) RetrievalPolicy, ConsumptionPolicy, LifetimePolicy + buffer maximum age (= combination of lifetime policies)

5.2 BUFFERED EVENT HANDLING

Why do we need a buffer to allow early event subscription?

What is the desired functionality of the event buffer? What functionality (API) does it expose? > this could be seen as an extension to the API that is exposed by a standard CEP Platform > standard platform API: registerQuery(queryString, notificationRecipient) : queryId, deleteQuery(queryId) > extended API: registerQuery(queryString): queryId, requestEvent(queryId, notificationRecipient), unsubscribe(queryId), deleteQuery(queryId)

5.3 EXTENDED PROCESS ENGINE BEHAVIOUR

=> As a link between the BPMN model and the Buffered Event handling

DECLARATION

Put your declaration here.

Potsdam, August 2017

Dennis Wolf

COLOPHON

This document was typeset using the typographical look-and-feel `classicthesis` developed by André Miede. The style was inspired by Robert Bringhurst's seminal book on typography "*The Elements of Typographic Style*". `classicthesis` is available for both LaTeX and LyX:

https://bitbucket.org/amiede/classicthesis/

Happy users of `classicthesis` usually send a real postcard to the author, a collection of postcards received so far is featured here:

http://postcards.miede.de/