

Modeling Test Cases in BPMN for Behavior-Driven Development (Zusammenfassung)

Daniel Lübke¹ Tammo van Lessen²

Abstract: Die Validierung von fachlichen und das Testen von ausführbaren Geschäftsprozessen ist eine herausfordernde Aufgabe in Prozessentwicklungsprojekten. Innerhalb des Terravis-Projekts, das einen Prozesshub für durchgehend digitalisierte Geschäftsprozesse für Schuldbrieftransaktionen zwischen Grundbüchern, Banken, Notaren und anderen Parteien entwickelt, wurde die Bewältigung dieser Aufgaben erfolgskritisch. Um die Kommunikation zwischen den Stakeholder zu verbessern und die Abdeckung durch automatisierte Testfälle zu erhöhen, wurde Behavior-Driven Development auf BPMN und Geschäftsprozesse adaptiert. Diese Technik wurde im Projekt erfolgreich eingeführt und durch alle Teammitglieder gut akzeptiert.

Keywords: BPMN, BDD, Executable Processes, Testing

1 Motivation

Im hier zusammengefassten Artikel [LvL16] wird beschrieben, wie Anforderungen und Tests analog Behavior-Driven Development (BDD) [No06] auch im Bereich ausführbarer Geschäftsprozesse, die in BPMN 2.0 [Ob11] modelliert sind, unterstützt werden können. Behavior-Driven Development ist von Test-Driven Development [Be03] und Domain-Driven Design [Ev03] inspiriert und versucht eine „Ubiquitous Language“ über Testfälle zu etablieren. Die Testfälle sollen in einer Art und Weise verfasst sein, damit sie sowohl von der fachlichen Seite verstanden – und damit validiert – werden können, als auch als ausführbare Testfälle interpretiert werden können.

Da Geschäftsprozesse oftmals in BPMN 2.0 modelliert werden, ist BPMN bereits die all-gemeingültige Sprache in Prozessentwicklungsprojekten, die von der Fach- und IT-Seite verstanden werden kann. BPMN 2.0 bietet ebenfalls eine Ausführungssemantik an, so dass Testfälle, die in BPMN modelliert wurden, ausgeführt werden können.

Das Projekt Terravis [BLM14] benötigte mit fortschreitender Komplexität und Größe der Prozesslösung [Lü15] verstärkt automatisierte Tests und eine Möglichkeit mit den unterschiedlichen Stakeholder, wie Grundbuchverwaltern, Bankprozessverantwortliche und Notare, Prozesse und ihre vielen Varianten zu besprechen und zu validieren. Ein weiterer Treiber war die Anforderung der gleichzeitigen Unterstützung von verschiedenen Service-versionen zu den verschiedenen Typen von Partnersystemen.

¹ innoQ Schweiz GmbH, Leibniz Universität Hannover/FG Software Engineering, Gewerbestr. 11, CH-6330 Cham, daniel.luebke@{innoq.com,inf.uni-hannover.de}

² innoQ Deutschland GmbH, Krischerstr. 100, 40789 Monheim am Rhein, tammo.van-lessen@innoq.com

2 Testmodelle & Generierung

Um Testfälle zu beschreiben, beschränken wir die Ausdrucksstärke der BPMN 2.0-Modelle so, dass die beschriebenen Testfälle deterministisch und möglichst sequentiell sind: Es sind keine Entscheidungskonstrukte erlaubt. Testnachrichten, die an den Prozess geschickt werden, sowie Assertions auf den vom Prozess gesendeten Nachrichten werden als natürlichsprachlicher, templatebasierter Text beschrieben. Die Templates enthalten Platzhalter, die dazu dienen, Werte in die Nachrichten sowie in die Assertions zu generieren. Damit ist eine Entkopplung der fachlichen Darstellung innerhalb der Testfälle und der technischen Konstrukte innerhalb der Templates, die in sogenannten Mappings definiert werden, gewährleistet. Zu den Mappings gehört ebenfalls eine Zuordnung der BPMN 2.0-Nachrichtenflüsse auf Webservice-Operationen und die jeweiligen Endpunkte. Somit sind alle technischen Informationen zur Testfallausführung und damit auch zur -automatisierung vorhanden.

Da Terravis bereits Unit-Tests für Prozesse mittels BPELUnit [Lü07] umgesetzt hat, wurde ein Generator geschrieben, der die BPMN 2.0-Testmodelle liest, die Template-Definitionen auswertet, diese mittels der Mappings in SOAP-Nachrichten und XPath-Assertions umwandelt und damit dann Testsuite generiert. Dabei liegen die Mappings als Tabellen extern von den Testmodellen, so dass dieselben Testmodelle mittels verschiedener Mappings verschiedene Serviceversionen testen können.

Literaturverzeichnis

- [Be03] Beck, Kent: Test-Driven Development by Example. Addison-Wesley, 2003.
- [BLM14] Berli, Walter; Lübke, Daniel; Möckli, Werner: Terravis – Large Scale Business Process Integration between Public and Private Partners. In (Plödereder, Erhard; Grunske, Lars; Schneider, Eric; Ull, Dominik, Hrsg.): Lecture Notes in Informatics (LNI), Proceedings INFORMATIK 2014. Jgg. P-232. Gesellschaft für Informatik e.V., Gesellschaft für Informatik e.V., S. 1075–1090, 2014.
- [Ev03] Evans, E.: Domain-Driven Design: Tackling Complexity in the Heart of Software. Addison-Wesley, 2003.
- [Lü07] Lübke, Daniel: Test and Analysis of Service-Oriented Systems. Springer, Kapitel Unit Testing BPEL Compositions, 2007.
- [Lü15] Lübke, Daniel: Using Metric Time Lines for Identifying Architecture Shortcomings in Process Execution Architectures. In: Software Architecture and Metrics (SAM), 2015 IEEE/ACM 2nd International Workshop on. IEEE, S. 55–58, 2015.
- [LvL16] Lübke, Daniel; van Lessen, Tammo: Modeling Test Cases in BPMN for Behavior-Driven Development. IEEE Software, 2016:17–23, Sep/Oct 2016.
- [No06] North, D.: Introducing BDD,. <http://dannorth.net/introducing-bdd>, 2006.
- [Ob11] Object Management Group: Business Process Model and Notation (BPMN) Version 2.0. Bericht, jan 2011.