Projektarbeit «SoMaDi»

*Kritische Würdigung*

von: Sonja Bachmann, Markus Rüttimann, Dirk Saleschus

Inhalt:

Vorwort

Die Projektgruppe «SoMaDi» befasste sich im Ramen einer Gruppenarbeit des Moduls 2 des CAS Solution Design 2017-2 mit der Prozessautomatisation einer Immobilienfirma. Die bestehenden Prozesse LOPK1 – 3 der Aufgabenstellung sollten in einem ersten Schritt in ein fachlich korrektes Prozessmodell überführt und danach ausführbar gemacht werden. Die Automatisierung sollte dabei mit der Entwicklungsmethode des *Rapid Prototyping* umgesetzt werden.

Projekt

Das Projekt inklusive Dokumentation ist auf github.com verfügbar: <https://github.com/Toomy/SL2017_ZHAW_LOPK>

LOPK1 (Sonja)

Prozess für die Bearbeitung von Vermietungen.

Einführung für Thomas (HowTo):

Der Prozess LOPK1 wurde in verschiedene Teile gegliedert. Die fachliche Modellierung sowie der Prototyp, welche jedoch nicht vollständig ausführbar ist. Im Fachmodell wurden zwei Rollen Vermieter und Mieter identifiziert. Im Prototypen findet sich jedoch nur die Rolle Vermieter wieder und die Prozesse des Mieters wurden in den Simulationsprozessen abgebildet.

Hier einige Erklärungen zum Protoypen:

* Der Prototyp startet mit einem Startevent bei dem das Vermietungsobjekt sowie die ID dazu eingetragen werden muss. Danach müssen gewisse Eckdaten zum Vermietungsobjekt erfasst werden.
* Nach der Publizierung des Mietangebotes wird auf eingehende Bewerbungen gewartet. Dies wurde mit einem Simulationsprozesse für Send/Empfang Events modeliert.
* Sobald eine Bewerbung emfangen wurde, wird diese auf Gültigkeit aufgrund einer Rule geprüft. Dies wurde mittels einem DMN modelliert.
* Der Unterprozess der eingehenden Bewerbungen ist ein laufender Prozess, welcher sich immer wieder wiederholt bis die Ablaufzeit eingetroffen ist.
* Danach erfolgen einige manuelle Schritte zur Auswahl des geeignetsten Mieters bis hin zum Versand des Vertrages.
* Der Erhalt des Vertrages wurde wiederum ausführbar modelliert, jedoch konnten hier einige Fehler zu den verschiedenen Gateways nicht gefunden und behoben werden. Gemäss Fehlermeldung sind gewisse SequenceFlows vorhanden, welche im Modell nicht mehr gefunden werden können. Dies führt dazu dass die Ausführbarkeit hier stoppt.
* Die Absage an andere interessierte Mieter erfolgt wiederum in einem manuellen Schritt.
* Der Abschnitt bezüglich Prüfung des Mietzinseingangs und dem anschliessenden Mahnprozess wurde wiederum ausführbar modelliert, kann jedoch aufgrund des vorhin beschriebenen Problemes nicht ausgeführt werden.
* Die letzten Schritte des Prototypen wie Mietfläche übergeben und Daten Mietverhältnis zusammenstellen wurden nicht ausführbar modelliert.

Fachmodell:

Die fachliche Modellierung wurde in einem seperatem Modell vorgenommen. Es trägt den Namen LOPK1\_Fachmodell.bpmn. Dieses Modell ist nicht ausführbar. Die fachliche Modellierung wurde vorgenommen um sich Schritt für Schritt in die BPMN Thematik einzuarbeiten.

Ausführbares Modell:

Das ausführbare Modell gliedert sich in verschiedene Teile:

Prototyp und Hauptmodell:

LOPK1\_Fachmodell.bpmn - Gesamt Vermietungsprozess

Simulationsprozesse:

LOPK1\_Bewerbungsdossier\_eingereicht.bpmn - Empfang von Bewerbungen von Mieter

LOPK1\_Absage senden.bpmn - Versand von Absagen an nich geeignete Mieter aufgrund der Bonität (DMN LOPK1\_BoniRule.dmn)

DMN:

LOPK1\_BoniRule.dmn - Entscheidungen bezüglich Überprüfung der Bonität von Bewerbern (Mieter) auf die Vermietung

In der Einführung wurde erklärt, dass der Prototyp leider nicht ausführbar ist und die Fehler, trotz stundenlanger Suche mit lieber Hilfe mit meinem Projektkollegen, nicht behoben werden konnten. Einie Abschnitte sind jedoch, wie oben beschrieben, ausführbar modelliert und können im BPMN Prozess gesichtet werden.

LOPK2 (Markus)

Prozess für die Bearbeitung von Mietvetragsänderungen.

Einführung für Thomas (HowTo):

Der Prototyp kann im Camunda aus der Taskliste mit «LOPK2 Prototype» gestartet werden.

Während der Ausführung des Prozesses wechselst du zwischen den Personas «Sachbearbeiter Mietvertragsänderung», «Mieter» und «Schlichtungsstelle» gemäss den erfolgten Antworten hin und her.

Du bekommst einen Usertask wenn der Prozess beendet ist, damit du sehen kannst was das Resultat war. Dies dient lediglich dem Verständnis.

Jeder User Task hat in der «generated Form» Hinweise was zu tun ist und welche Antworten zur Verfügung stehen.

Die Datei *LOPK2\_Prototype.bpmn* hat zu jedem Task eine Erklärung in Form einer Annotation. Damit kannst du meinen Überlegungen folgen und andere, am Prototyp beteiligte, können sich schnell einarbeiten.

Fachmodell:

Die Datei *LOPK2\_Fachmodell.bpmn* enthält das Fachmodell. Dieses Modell ist nicht ausführbar.  
Die Datei *LOPK2\_DataModell.pfd* enthält das Datenmodell dieses Prozesses. Es ist auch im Anhang enthalten.

Ausführbares Modell:

Der *LOPK2\_Prototype Prozess* kann gestartet werden.

*LOPK2\_Customer\_Sim.bpmn* Simulationsprozess des Mieters.

*LOPK2\_Schlichtungsstelle\_Sim.bpmn* Simulationsprozess der Schlichtungsstelle.

*LOPK2\_Contract\_Change.cmmn* Das Case Management zum Bearbeiten der einzelnen Vertragsänderungen. Ist im Zustand wie wir es «Aborted» haben. Die html Formulare habe ich im Netbeans erstellt.

*LOPK2\_LetterResponseForkRule.dmn* Die DMN Matrix für die Entscheidung bezüglich der Antwort des Mieters.

Stärken / Schwächen der Modellierten Lösung

Die wesentlichen Aspekte der Aufgabenstellung wurden umgesetzt, es fehlt nichts.

Ich habe mich im speziellen dem Aspekt der «Verfolgbarkeit und Dokumentation» des Prozesses gewidmet. Ich finde es wichtig, dass in diesem Fall die gesamte Kommunikation und Korrespondenz erfasst und gespeichert wird. Dies habe ich im Prozessmodell 100% umgesetzt.

Wegen der Übersichtlichkeit (Nur eine Persona, Sachbearbeiter) habe ich den ganzen Prozess in nur einer Lane modelliert. Es gibt also lediglich die Prozess Lane. Dies sollte in einer richtigen Simulation besser gemacht werden um Unklarheiten mit dem Business Analysten zu vermeiden.

LOPK3 (Dirk)

Prozess für die Bearbeitung der Kündigung von Mietverträgen

Einführung für Thomas (HowTo):

Für die fachliche Modellierung wurden drei verschiedene Rollen (*Actors*) identifiziert, die als unabhängig voneinander eingestuft wurden, jeweils eigene Prozesse mit nur einem ausführlich modellierten Pool haben und miteinander kommunizieren: Der Vermieter, der Mieter und ein Richter.

Die Modellierung der drei Prozesse ist in folgenden Dateien abgelegt:

* Vermieter: LOPK-3\_Prototyp.bpmn
* Mieter: LOPK-3\_Prototyp\_dummyRenterMessageModeler.bpmn
* Richter: LOPK-3\_Prototyp\_RichterBestimmtAusweisungDesMieters.bpmn

Nur der Prozess des Vermieters wurde ausführlich modelliert – dies ist der eigentliche Protoyp. Die anderen beiden Prozesse sind Platzhalter mit minimaler Modellierung.

Der Prozess des Vermieters hat zwei verschiedene Startereignisse und damit zwei Möglichkeiten, im Camunda-Modeler über die Camunda-Taskliste gestartet zu werden. Der Vermieter kann den Prozess direkt durch „LOPK3-Vermieter“ starten. Der Mieter kann den Prozess „LOPK3\_dummyRenterMessageModeler“ starten, worauf durch ein Sendeereignis der Prozess „LOPK3-Vermieter“ gestartet wird. In beiden Fällen liegt der Fokus auf den Aktivitäten des Vermieters.

Die Annotationen, Kommentare und Formen im Prozess „LOPK3-Vermieter“ sollen das Verständnis des Prozessablaufs erhöhen.

Fachmodell:

Es wurde auf eine separate Modellierung von Fachmodell und ausführbarem Modell (Prototyp) verzichtet. Statt dessen gibt nur einen ausführbaren Prototypen. Der Grund ist, dass von Anfang an der Fokus auf die Lauffähigkeit des Prototypen gelegt und der Prozess des Vermieters daher von Beginn an ausführlich modelliert wurde.

Ausführbares Modell:

*LOPK-3\_Prototyp.bpmn* Der Prototyp kann über den Prozess LOPK3-Vermieterkann gestartet werden.

*LOPK-3\_Prototyp\_dummyRenterMessageModeler.bpmn* Simulationsprozess des Mieters, welcher den Prozess LOPK3-Vermieterstartet. Ist nur als Platzhalter vorgesehen. Das zweite Sendeereignis wird vom Prozess „LOPK3-Vermieter“ nicht erkannt.

LOPK-3\_Prototyp\_RichterBestimmtAusweisungDesMieters.bpmn Simulationsprozess des Richters, welcher für den Prozess „LOPK3-Vermieter“ ein Signal schicken kann. Ist nur als Platzhalter vorgesehen. Das Signalsendeereignis wird vom Prozess „LOPK3-Vermieter“ nicht erkannt.

Stärken / Schwächen der Modellierten Lösung

Alle Aufgaben und Fallunterscheidungen der Prozessbeschreibung wurden berücksichtigt. Als Stärke wird hier aufgefasst, dass zusätzlich Ausnahmefälle und sinnvolle Zusatzaufgaben modelliert wurden, die ein realistischeres und vollständigeres Bild des Prozesses liefern.

Eine weitere Stärke wird in der Nutzung von Zufallszahlen gesehen, um unterschiedliche Situationen nicht über den Einsatz von Nutzereingaben sondern über Ereigniswahrscheinlichkeiten zu modellieren. Somit konnte eine schnellere Lauffähigkeit des Prototypen erreicht werden. Ein Beispiel ist der Skript-Task „Mieterdossier aus CRM laden“. Hier wird als Resultat eine statistische Zufallsvariable mit Standardnormalverteilung generiert. Diese soll Ereigniswahrscheinlichkeiten simulieren (eine Binomialverteilung wäre natürlich sinnvoller, aber das Prinzip ist ähnlich). Der Wert wird im folgenden Gateway benutzt, um die Abzweigung zu ermitteln. Dabei wurde versucht, die Verteilung der Ereigniswahrscheinlichkeiten realistisch einzuschätzen. Im Beispiel wurde angenommen, dass in der Mehrheit der Fälle die Stammdaten gut gepflegt sind, in einem nicht zu kleinen Teil der Fälle jedoch Inkonsistenzen vorhanden sind (dies trifft z.B. auf Firmen zu, die nicht sehr standardisierte und sichere Datenerfassungsprozesse haben).

Die grössten Schwächen liegen in der unvollständigen Lauffähigkeit (durch mangelhafte Implementierung des Nachrichtenaustauschs) und der teilweise inadäquaten Modellierung. Dies ist dem fehlenden Verständnis der technischen Modellierung mit Camunda geschuldet. Es konnten weder CMMN, noch DMN oder eine durchgehende Kommunikation über Nachrichten modelliert werden. Ein durchgängiger Prozessablauf ist nur für den Happy Path, nicht jedoch für alle möglichen Sonderfälle möglich.

CMMN wäre nach dem Skript-Task „Mieterdossier aus CRM“ laden sinnvoll gewesen, um die manuelle und nur schwach strukturierten Aufgaben der manuellen Aufräumarbeit zu modellieren. Statt dessen wurde ein parallele Ausführung mehrere Tasks gewählt.

DMN wäre ein sinnvollerer Ersatz des Skript-Tasks „Fristen“ prüfen gewesen, da so die verschiedenen Berechnungen des Kündigungstermins hätten berechnet werden können. Trotz der Vorlagen von Thomas Keller und Markus Rüttimann ist es nicht gelungen, das DMN erfolgreich zu implementieren.

Der Prototyp verfügt nicht über die Möglichkeit einer vollständigen Kommunikation über Nachrichten. Das Startereignis „Kündigung vom Mieter erhalten“ ist ein seltenes funktionierendes Beispiel. Obwohl alle folgenden Nachrichten auf dieselbe Weise modelliert wurden, meldet Camunda einen fehlenden „correlation key“ und beendet den Prozess (allerdings nicht über einen Fehler). Aus dem Grund wurde die weitere Modellierung der Prozesse des Mieters und des Richters aufgegeben.

Das zweimalige Vorkommen des Data Stores „CRM“ ist unschön, wurde aber eingesetzt, um die Datenströme übersichtlicher modellieren zu können.

Umsysteme

Damit eine Automatisation der Prozesse vorgenommen werden kann, müssen Annahmen über die Umsysteme gemacht werden. Wir haben die folgenden Umsysteme definiert:

System zum Senden und Empfangen von Post:

**Versenden:** Es existiert eine API über welche Briefe verschickt werden können. Die API nimmt eine Adresse und ein elektronisches Dokument entgegen. Das angeschlossene System druckt den Brief inklusive Couvert aus und verpackt den Brief und verschickt ihn automatisch.

**Empfangen:** Es existiert ein externes System in welchem eingehende Post verarbeitet wird. Die Briefe werden elektronisch gescannt und anhand der vorhandenen Referenznummern automatisch oder manuell ins LOPK System eingespeist. Die Einspeisung sorgt dafür, dass die entsprechenden Prozess Instanzen benachrichtigt werden. Das externe System sendet diese Meldung

System CRM System:

Es existiert ein System in welchem sämtliche Kundeninformationen gespeichert sind. Zu diesen Informationen gehören die Stammdaten (Name, Adresse, etc.) aber auch die vollständige Dokumentation der Kommunikation mit dem Kunden sowie die Dokumentation allfälliger Schlichtungsentscheide.

System: Facility Management System:

Dieses System enthält sämtliche Informationen aller Liegenschaften und Wohnungen wie Gösse, Anzahl Zimmer, Spezifische Einrichtungen und Ausbauten.

Abbildung von Umsystemen:

Die Umsysteme wurden als Dummy Prozesse oder mit «Script tasks» modelliert. Die Dummy Prozesse kommunizieren mit Messages mit dem Prozess.

Schwierigkeiten Lessions learned.

Es hat sich gezeigt, dass bestehende Prozesse vor dem Hintergrund einer Automatisation von ungenügender Qualität sind. Die Aussage, dass «nicht automatisierte Prozesse die Dokumentation einer Scheinwelt sind», hat sich mehrfach bestätigt.

Modellierung

Damit die Prozesse automatisiert werden können, müssen diese gemäss dem «BPMN Lifecycle» Re-Engineered werden. Dabei müssen sowohl die Toolchain, die Benutzer sowie die Unternehmensarchitektur berücksichtigt werden.

Überführung vom vorhandenen Prozess Modell in ein fachlich korrektes Modell.

* Die vorgefundenen Prozesse sind lückenhaft und nicht Vollständig.
* Impliziertes wissen ist notwendig damit der Prozess überhaupt funktioniert.

Technische Schwierigkeiten.

Wir hatten viele Schwierigkeiten, vor allem weil uns der Technologie Stack unbekannt ist und teilweise Programierkenntnissere notwendig sind welche nicht bei allen vorhanden sind.

Allerdings hatte auch der Technologie stack probleme, nämlich:

CMMN

Wir haben den CMMN modelliert, aber nicht zum laufen gebracht. Die CMMN Prozess Instanz hat sich immer vorzeitig beendet und dadurch ist der ganze Prozess zum stehen gekommen. Wir haben den Fehler nicht gefunden. Der entsprechende Task wurde mit einem Kommentar versehen, jedoch aus Frust nicht neu im BPMN Modelliert, weil die nächstbeste Modellierungsmöglichkeit des Ad-Hock Unterprozesses von Camunda nicht unterstützt wird.

DMN

Wir haben viel Zeit mit einer Fehlermeldung bezüglich zweideutiger IDs verloren. Camunda bietet hier keine Unterstützung, und ist einmal ein Fehler drinnen kommt man fast nicht auf die Lösung.

Überführung vom fachlichen Modell in ein ausführbares Modell.

* Technische Randbedingungen vom Technologie Stack sind nicht vollständig bekannt. Dass der zum Bsp. «correlationKey» genau so heissen muss und mit einem start-script gesetzt werden muss ist schwierig herauszufinden.

Fazit

* Der anspruchsvolle Technologies Stack «Camunda» stellten jeden Teilnehmer wiederholt vor grosse Herausforderungen.
* Die für ein ansprechendes Prototyping erforderlichen Programmierkenntnisse sind nicht im erforderlichen Masse bei in der Gruppe vorhanden.
* Die Aufgabenstellung hat die Dimensionen der bei einer Prozessautomatisation zu überwindenden Probleme hervorragend illustriert.
* Wir haben im Rahmen der Projektarbeit ein gutes Verständnis über An- und Herausforderungen einer Prozessautomatisation bekommen und können diese in der Praxis anwenden -allerdings nicht in der programmiertechnischen Umsetzung, wohl aber in der Konzeption und Projektleitung.

Anhang:

ERD LOKP2 Prozess

