

Flexible Prozessautomatisierung durch Integration der Prozess-Engine Camunda in eine generisch konfigurierbare Stammdatenapplikation

Prof. Dr. Peter Edelmann

Prof. Dr. Armin
Wagenknecht

M.Sc. Michel Jonathan
Schmitz

Technische Hochschule
Mittelhessen
Wilhelm-Leuschner-Straße 13
61169 Friedberg
Peter.Edelman@mnd.thm.de

Technische Hochschule
Mittelhessen
Wiesenstraße 14
35396 Gießent
Armin.Wagen-
knecht@mni.thm.de

Technische Hochschule
Mittelhessen
Wiesenstraße 14
35396 Gießen
Michel.Jonathan.
Schmitz@jts.thm.de

ABSTRACT

Bei komplexen Verwaltungsprozessen ist die Vielfalt möglicher Prozessabläufe oft nicht vorhersehbar oder eine vollständige Modellierung nicht wirtschaftlich. WissensarbeiterInnen benötigen Möglichkeiten, um auf Abweichungen vom definierten Prozessablauf reagieren zu können. Oft werden daher neben dem Workflow-Management-System separate Applikationen zur Stammdatenverwaltung sowie zur Durchführung der nicht modellierten Prozessabläufe erstellt. Die vorliegende Arbeit untersucht die Hypothese, dass die Flexibilität für WissensarbeiterInnen durch Integration einer Prozess-Engine (Produkt Camunda) in eine generisch konfigurierbare NoSQL-Neo4J-basierte Stammdatenapplikation erhöht werden kann, durch Entwicklung eines Prototyps. In diesem wird die Prozess-Engine ausschließlich auf die Kontrolle des Workflows reduziert. Die Engine delegiert die Bearbeitung der im Prozess angefallenen Daten als BenutzerAufgabe an die Stammdatenapplikation und schränkt dabei den Aktionsraum der WissensarbeiterInnen durch Rechte ein. Die Integration bietet WissensarbeiterInnen einerseits die Unterstützung durch BPMN-definierte Workflows und andererseits die Flexibilität einer manuell bedienbaren Stammdatenapplikation. Durch das vorliegende Verfahren kann etwa ein nicht modellierter Change-Prozess von WissensarbeiterInnen manuell ausgeführt werden.

SCHLÜSSELWÖRTER

Prozessautomatisierung, Prozessflexibilisierung

EINLEITUNG

Die Prozessmodellierung kämpft mit dem Problem, die Balance zwischen Flexibilität und Strukturiertheit von Prozessabläufen zu finden.

Die Modellierung aller Ausführungsalternativen für die komplexen Prozesse eines Büroumfeldes in einem Workflow-Management-Systeme ist oft wegen des hohen Aufwands nicht sinnvoll.

WissensarbeiterInnen (Personen, deren Aufgabe es ist, Wissen zu erzeugen, zu verbreiten und anzuwenden (Davenport 2005, S. 10) benötigen vielfältige Möglichkeiten, um auf Abweichungen vom definierten Prozessablauf reagieren zu können.

Oft werden daher neben dem Workflow-Management-System separate Applikationen zur Stammdatenverwaltung sowie zur Durchführung der nicht modellierten Aktivitätsflüsse erstellt.

Die vorliegende Arbeit untersucht, ob die Flexibilität für WissensarbeiterInnen durch Integration eines Workflow-Management-Systems in eine generisch konfigurierbare Stammdatenapplikation erhöht werden kann.

Die Integration könnte einerseits die Unterstützung durch BPMN definierte Workflows bieten und andererseits die Flexibilität einer manuell bedienbaren Applikation, welche beliebige Aktivitätsflüsse innerhalb des durch das zugehörige Rechtesystem eingeschränkten Aktionsraums erlaubt.

STAND DER FORSCHUNG

Für die Modellierung von Prozessdefinitionen existieren zwei verschiedene Paradigmen.

Bei der imperativen Prozessmodellierung (Fahland et al. 2009, S. 2)) werden alle Ausführungsmöglichkeiten und Zustände des Modells definiert. Im Falle einer neuen Alternative, muss diese explizit dem Modell hinzugefügt werden (Pescic 2008, S. 80). Fahland stellt die Frage, ob Workflow-Management-Systeme, die diese Art der imperativen Prozesse unterstützen, auch im Büroumfeld angewendet werden können und wirft dabei das Problem auf, dass Prozessbeteiligte im Büro mehr Möglichkeiten benötigen, um auf Abweichungen vom definierten Prozessablauf reagieren zu können (Reijers et al. 2003, S. 1-2).

Der Wunsch nach Flexibilität wird durch die deklarative Modellierung in den Vordergrund gerückt (van der Aalst et al. 2009, S. 99-100). Bei der deklarativen Prozessmodellierung werden Einschränkungen definiert, über die

implizit die möglichen Handlungsalternativen bestimmt werden. Das Hinzufügen von Einschränkungen bedeutet, dass Handlungsalternativen ausgeblendet werden (Pesic 2008, S. 80). Durch die wählbaren Handlungsalternativen können WissensarbeiterInnen den Prozess in einem gewissen Rahmen frei bearbeiten. Durch den Fokus auf Handlungsalternativen fehlt allerdings die Unterstützungsfunktion des Workflow-Management-Systems. Weniger gut ausgebildete Prozessbeteiligte können daher nicht effizient mit dem System arbeiten. Es entsteht dabei das Problem, die richtige Balance zwischen Flexibilität und Strukturiertheit von Prozessen zu finden (Ter Hofstede et al. 2009, S. 175-178; van der Aalst et al. 2009, S. 99-100).

Zu beiden Ansätzen existieren mittlerweile etablierte Standards der OMG. Die BPMN ermöglicht die imperative Prozessmodellierung (OMG, 2013) und Case Management Model and Notation (CMMN) die deklarative Prozessmodellierung (OMG 2016)

Bei dem Versuch eine Lösung für das Balance-Problem zu finden, gehen AutorInnen unterschiedliche Wege. Unter anderem ist die Hybridlösung BPMN-D entstanden. Die BPMN wird dabei um deklarative Elemente erweitert. Dazu wird ein Konvertierungsalgorithmus vorgestellt, der aus der deklarativen, einschränkungs-basierten Modellierungssprache Declare ein gültiges BPMN-D Modell erzeugt (De Giacomo et al. 2015, S. 88-99). Ein anderer Vorschlag, um die Vorteile der imperativen und deklarativen Modellierung zu nutzen, ist die Kombination der BPMN und CMMN (Hinkelmann and Pierfranceschi 2014, S. 10).

VORBEMERKUNG

Problemschilderungen, Konzepte, Szenarien und Prototyp orientieren sich zur besseren Verständlichkeit und im Hinblick auf eine gewünschte Praxisnähe an dem im Fachbereich MND der Technischen Hochschule Mittelhessen betriebenen Onlinesystem für den Verwaltungsprozess zur Abwicklung einer Abschlussarbeit, z.B. Masterarbeit, im Studiengang Wirtschaftsinformatik.

PROBLEME HERKÖMMLICHER PROZESSAUTOMATISIERUNGSSYSTEME

Geringe Flexibilität bei der Bearbeitung von

HumanTasks

Herkömmliche Prozessautomatisierungen mit weit verbreiteten Prozess-Engines, z.B. dem Produkt Camunda, gewähren der AnwenderIn bei Ausführung einer Benutzeraufgabe (HumanTask) nur eine geringe Flexibilität.

So kann zum Beispiel der Prüfungsausschussvorsitz im Rahmen der Genehmigung einer Masterarbeit nur Thema und Stammdaten der StudentIn einsehen. Die besuchten Wahlpflichtfächer, die Themen vorhergehender Seminare, Praktika, oder Informationen über die betreuende Firma sind nicht verfügbar.

Möchte die ProzessadministratorIn dem Prüfungsausschussvorsitz für die betrachtete Benutzeraufgabe eine größere Flexibilität einräumen, so muss der Prozess mit

verschiedenen AutomaticTasks zur Datenbeschaffung und weiteren Webseiten (HumanTaskViews) zur Datendarstellung erweitert werden. Eine solche Erweiterung kann in der Regel nur von ProgrammierInnen mit Erfahrungen in den Bereichen BPMN, Webseiten, und Datenbanken vorgenommen werden.

Separate Applikationen zur Durchführung der im Prozess nicht modellierten Aktivitätsflüsse

Prozesse in der realen Bürowelt sind häufig sehr komplex. Bedingt durch diese hohe Komplexität werden in der Praxis oft nicht alle erforderlichen Datenmanipulationen und Entscheidungsmöglichkeiten durch zugehörige Pfade der Prozesse bzw. Subprozesse abgebildet. Im betrachteten Beispiel der Masterarbeit mag etwa kein Change-Prozess integriert sein, der eine nachträgliche Änderung des Themas oder eine Verschiebung des Abgabedatums wegen Krankheit abbildet.

Auch ein Bedienfehler, beispielsweise das Absenden fehlerhafter Eingaben, kann oft nicht kompensiert werden. Die im Rahmen der HumanTask von der Prozess-Engine zur Verfügung gestellten Webseiten (HumanTaskViews) bilden die einzige Möglichkeit für die BenutzerIn, Daten anzusehen und zu editieren.

Daher existieren oft neben dem System zur Prozessautomatisierung ein oder mehrere Applikationen zur Verwaltung der im Rahmen des Prozesses anfallenden Stammdaten. Diese Applikationen erlauben es den WissensarbeiterInnen, nicht modellierte Aktivitätsflüsse im Rahmen der ihr erteilten Rechte und basierend auf ihrem Prozessverständnis manuell auszuführen.

Dazu gehören etwa eine Korrektur der Stammdaten außerhalb des Prozesses, eine Bearbeitung der Stammdaten im Falle eines Prozessabbruchs oder eines nicht modellierten Prozesspfades.

FLEXIBLE PROZESSAUTOMATISIERUNG DURCH INTEGRATION DER PROZESS-ENGINE CAMUNDA IN EINE GENERISCH KONFIGURIERBARE STAMMDATEN- APPLIKATION

Der vorliegende Ansatz zielt darauf ab, die oben genannten Probleme durch eine Kopplung der Prozess-Engine mit einer generisch konfigurierbaren Stammdatenapplikation zu reduzieren.

Dabei übernimmt jede der beiden Komponenten, Prozess-Engine und Stammdatenapplikation genau die Aufgaben für die sie besonders geeignet ist und konstruiert wurde. D.h. die Stammdatenapplikation beschafft und präsentiert der AnwenderIn im Rahmen einer HumanTask die gewünschten Daten und die Prozess-Engine kontrolliert den Workflow (vgl. Abbildung 1).

Die imperative BPMN-Beschreibung des durch die Prozess-Engine abzuarbeitenden Workflows wird mit der

deklarativen Beschreibung des Aktionsraumes einer AnwenderIn durch Rollen und Rechte in der Stammdatenapplikation kombiniert.

Die Webseite kann durch die Verzahnung mit der Prozess-Engine nur im Rahmen der HumanTask verwendet werden, nicht aber, um nachträglich z.B. das Thema oder

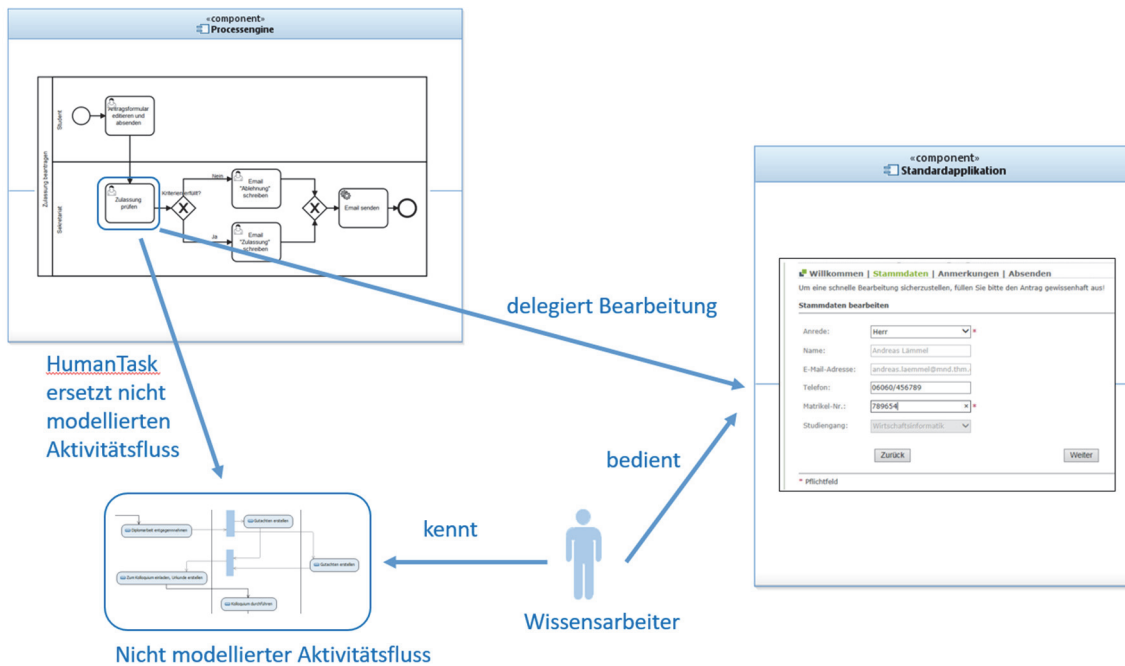


Abbildung 1: Delegation Aktivitätsfluss an Wissensarbeiter mittels HumanTask

FACHLICHE ARCHITEKTUR DES PROTOTYPS

Reduktion der Prozess-Engine auf die Workflow

Steuerung

Das Kommunikationsdiagramm in Abbildung 2 zeigt Objekte und Methodenaufrufe zur Darstellung der Kollaboration für eine herkömmliche Verwendung einer Prozess-Engine (grüne Objekte sind der Domäne Stammdatenverwaltung, blaue Objekte sind der Domäne Prozess-Engine zuzuordnen).

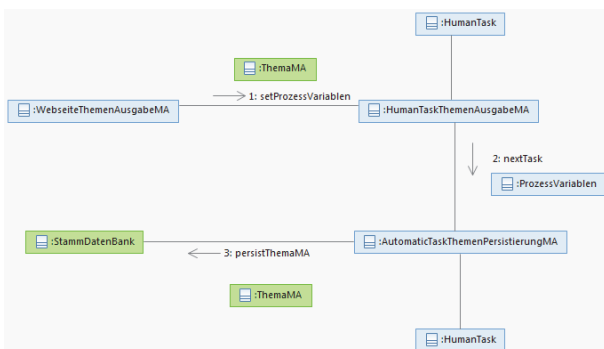


Abbildung 2: Kopplung der Web-Seiten und des Daten-Transports an die Prozesssteuerung

Die Angabe des Themas der Masterarbeit erfolgt im HumanTaskView also in der der HumanTask zugeordneten Webseite. Das Thema wird in die Prozessvariablen der Prozess-Engine transportiert und in der nachfolgenden AutomaticTask in die Stammdatenbank persistiert.

den Abgabezeitpunkt zu korrigieren.

Nach einem Abbruch des Prozesses kann aufgrund der Kopplung mit der Prozess-Engine keine den Human-Tasks zugeordnete Webseite bedient werden.

Abbildung 3 zeigt die Entkopplung der Webseiten und des Daten-Transports von der Prozesssteuerung (vgl. Abbildung 2 zur Erläuterung des Kommunikationsdiagramms).

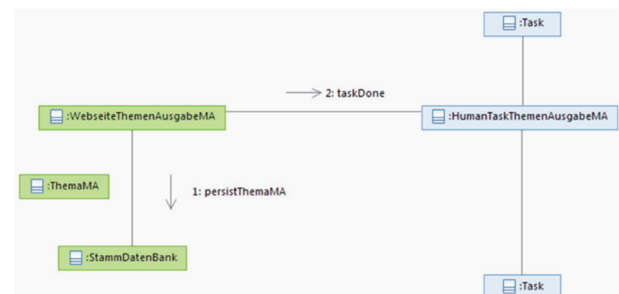


Abbildung 3: Entkopplung der Web-Seiten und des Daten-Transports von der Prozesssteuerung

Die Angabe des Themas der Masterarbeit und die Persistierung in die Stammdatenbank erfolgen durch Bedienen der Stammdatenapplikation.

Die Aufgabe der angekoppelten Prozess-Engine reduziert sich auf die Steuerung der Aktivitäten der im Prozess Masterarbeit beteiligten AkteurInnen.

So kann zum Beispiel eine Änderung des Themas oder Abgabezeitpunktes ohne Beteiligung der Engine direkt in der Stammdatenapplikation erfolgen. Ggf. kann die AkteurIn innerhalb des durch das Rechtssystem einge-

schränkten Aktionsraums einen nicht modellierten Aktionsfluss manuell ohne Engine ausführen. Nach einem Ausfall der Prozess-Engine kann der Prozess ebenfalls manuell ohne Engine weitergeführt werden.

Abbildung 4 zeigt den Unterschied zwischen den Mechanismen einer herkömmlichen Prozess-Engine und dem vorliegenden Ansatz basierend auf der losen Kopplung im direkten Vergleich (vgl. Abbildung 2 zur Erläuterung der Kommunikationsdiagramme).

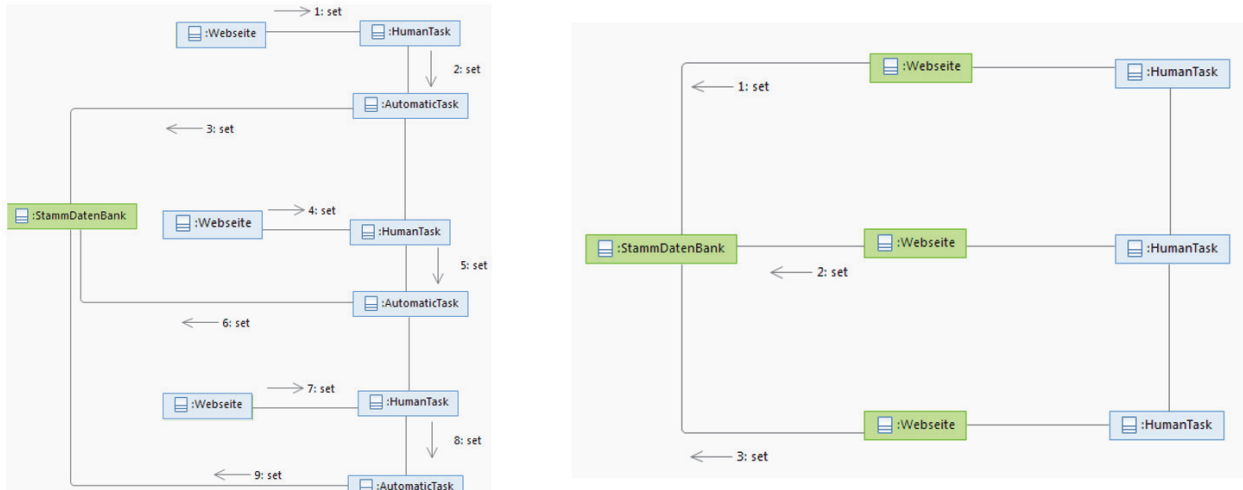


Abbildung 4: Unterschied Mechanismen bei enger Kopplung (links) und loser Kopplung (rechts)

Die Webseiten sind bei loser Kopplung vollständig in der manuell von der WissensarbeiterIn bedienbaren Stammdatenapplikation integriert und lediglich für die Workflow-Steuerung mit der Prozess-Engine gekoppelt.

Generisch konfigurierbare Stammdatenapplikation basierend auf NoSQL-Metadaten

Im Hinblick auf eine verbesserte Flexibilität für die ProzessadministratorIn wurde die Prozess-Engine an eine generisch konfigurierbare Stammdatenapplikation gekoppelt. Die Stammdatenapplikation unterstützt Listen- und Detailansichten zum Erzeugen, Lesen, Ändern und Löschen (CRUD) von generisch konfigurierbaren Entitäten. Die Zugriffssteuerung erfolgt einheitlich über Rollen, Rechte und Filter. Diese spezifizieren, die für die AnwenderIn sichtbaren Entitäten und Navigationsmöglichkeiten. Basis der Applikation bildet eine Neo4j-NoSQL-Datenbank. Diese unterstützt die Abspeicherung von Stammdaten in Form von gelabelten Nodes und Relations. Sie liefert Attribute und Relations in Form eines Arrays von String basierten Key-Value Pärchen. Die zur Laufzeit erzeugten Ansichten basieren auf Metadaten zu den Entitätstypen. Die Einschränkung der AnwenderIn und Validierung der eingegebenen Stammdaten erfolgt durch die Definition von MetaNodes. Diese legen für einen Entitätstyp die Bezeichnungen, Datentypen und Werteschränkungen der Attribute sowie der Relationen zu anderen Entitätstypen fest. Die Modellie-

rung von Rechten und Filtern in den Metadaten ermöglicht die Einschränkung des Zugriffs mittels der Neo4j Cypher Query Language basierend auf Graph Patterns bestehend aus Anker und Pfad. Die generische Stammdatenapplikation unterstützt eine anwendungsspezifische Codierung über konfigurierbare „SpecialServices“. Sie erlaubt z.B. das „Claimen“ („Claimen“ bezeichnet die Aktion, das eine NutzerIn eine Aufgabe für sich zur Bearbeitung beansprucht) einer HumanTask, ListItems mit speziellen Informationen anzureichern, allen Personen

einer Liste Emails zu senden oder die Generierung eines spezifischen Reports. Dazu werden im Meta-Modell alle anwendungsspezifisch zu codierenden Aktionen durch MetaActions modelliert und durch verschiedene Relations, z.B. forwardToViewAction, forwardToServiceAction, onCreateAction, mit der betroffenen MetaNode verbunden. Die MetaAction spezifiziert den SpecialService bzw den SpecialView. Die GUI-Handler delegieren Requests für MetaNodes ggf. an zugeordnete SpecialServices.

Repräsentanten der prozessrelevanten Informationen in der Stammdatenapplikation

Im Falle einer Prozessbearbeitung verwaltet die Prozess-Engine meist eine sogenannte Prozessvariable, welche alle prozessrelevanten fachlichen Informationen kapselt. Da diese Informationen in der vorliegenden Arbeit als Neo4j-Nodes in der Stammdatenapplikation verwaltet werden, wird zu jedem Prozess eine Metadaten-Struktur editiert, welche die prozessrelevanten fachlichen Informationen in Form von Entitäten repräsentiert. Diese Datenstruktur beinhaltet immer eine ProcessAggregat-Node, welche alle prozessrelevanten Entitäten aggregiert. Im Beispiel des Prozesses zur Verwaltung der Masterarbeit werden etwa Nodes zu dem Studierenden, der Firma, der BetreuerIn und zur Spezifikation von Thema und Bearbeitungszeitraum der Masterarbeit sowie eine aggregierende MasterArbeitProcessAggregat-Node erzeugt (vgl. Abbildung 5).

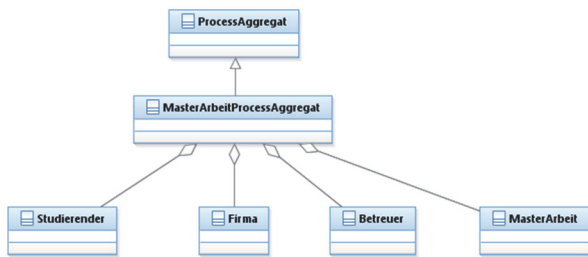


Abbildung 5: Repräsentanten der fachlichen Prozessinformationen

Repräsentanten des Prozesses und der HumanTasks in der Stammdatenapplikation

Damit eine AktuerIn eine HumanTask im Rahmen der Stammdatenapplikation „claimen“ kann, wird ein Repräsentant des von der Prozess-Engine erzeugten internen Camunda Prozesses und der zugehörigen zu bearbeitenden HumanTask in der Stammdatenapplikation benötigt. Daher wurden die entsprechenden Entitäten EngineProzess-Node und EngineHumanTask-Node in der Metadaten Struktur spezifiziert.

Beide Entitäten repräsentieren lediglich Elemente der Camunda Prozess-Engine und sind zu unterscheiden von den Entitäten, welche prozessrelevante fachliche Informationen kapseln.

Im Beispiel des Prozesses zur Verwaltung der Masterarbeit wird für den gestarteten Prozess eine EngineProzess-Node und für die auszuführende HumanTask „Masterarbeit Themenausgabe“ eine EngineHumanTask-Node instanziiert (vgl. Abbildung 6).

Die EngineProzess-Node und EngineHumanTask-Node enthalten die entsprechenden IDs der internen Elemente der Camunda-Prozess-Engine.

Die EngineProzess-Node aggregiert die zugehörigen EngineHumanTask-Nodes.

Die EngineProzess-Node referenziert die zugehörige ProcessAggregat-Node. Diese Referenz repräsentiert die Verbindung von der Prozesswelt der Engine zur Stammdatenwelt.

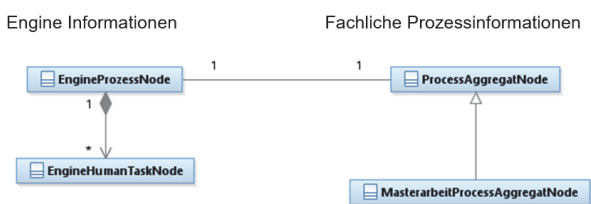


Abbildung 6: Prozess-Repräsentanten

Die Camunda-Prozess-Engine referenziert die EngineProzess-Node mittels Neo4J-ID im Rahmen der Prozessvariablen und das Interface zur Ansprache der Stammdatenapplikation über eine URL (REST Calls).

Rechte und Rollen

Task-Rollen sind temporäre Rollen, die nur zu dem Zeitpunkt aktiviert werden, wenn die BenutzerIn eine HumanTask bearbeitet. Der Name einer Task-Rolle ist per

Konvention identisch mit der Element-ID des korrespondierenden BPMN-Elements. Die Rolle beschreibt den Aktionsraum, den eine BenutzerIn im Rahmen der HumanTask verwenden können. Die BenutzerIn kann die Aufgabe innerhalb des vorgegebenen Aktionsraums, bearbeiten und ist dabei nicht auf feste Reihenfolgen von Ansichten angewiesen. Sie ist jedoch auf die für ihre Aufgabe relevanten Daten eingeschränkt. BenutzerInnen, die eine Lane-Rolle besitzen, sind autorisiert alle Aufgaben einer BPMN-Lane eines Prozesses zu bearbeiten. Eine Lane-Rolle kann Personengruppen zugeordnet werden. Die Zuordnung zu den Aufgaben der Lane erfolgt über einen Filter, der alle Benutzer-Aufgaben einer BPMN-Lane basierend auf Benennungskonventionen auffindet.

Einer eingeloggten AnwenderIn (UserIn) ist eine Role-Node zugeordnet und dieser wiederum mehrere Right-Nodes. Jede Right-Node aggregiert eine CRUDE-Specification für ein Node Label und Filter zur Selektion der Nodes des angegebenen Labels. CRUDE steht für create, read, update, delete, execute. Die Abbildung 7 zeigt die zugehörige Metadaten-Struktur.

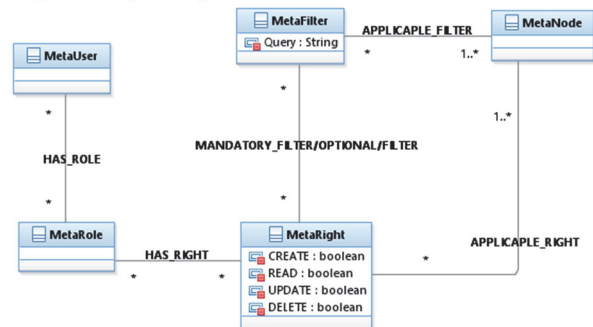


Abbildung 7: Rechte, Rollen und Filter

Filter werden mittels Neo4J-Cypher-Queries spezifiziert. Zu diesem Zweck muss der Startknoten der Query in Form der AnchorNode spezifiziert werden.

Durch die Abhängigkeit der Abfrage von der eingesetzten AnchorNode entstehen verschiedene Ergebnismengen.

Für die Stammdatenapplikation wird in der Regel die eingeloggte AnwenderIn als Startknoten verwendet.

Da verschiedene AnwenderInnen auf dieselben Rollen verweisen, Rollen also nicht User-spezifisch definiert sind, muss die AnchorNode User-spezifisch gesetzt werden.

Daher ist die AnchorNode eine Association-Klasse für die Association-User zur Rolle vgl. Abbildung 8.

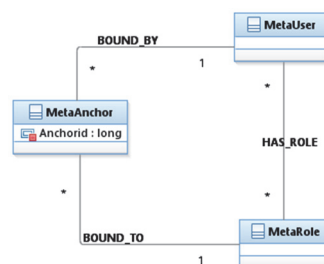


Abbildung 8: AnchorNode Metadaten-Struktur

Die AnchorNode kann erst zur Laufzeit gesetzt werden. Per Default wird für die Stammdatenapplikation die AnchorNode beim Einloggen vom Login-Controller im Rahmen der Rechte Zuweisung auf die Node-ID der eingeloggten AnwenderIn gesetzt. Im Falle einer Prozessbearbeitung wird in der Regel die ProcessAggregat-Node als AnchorNode verwendet (vgl. Abbildung 9). Bei Annahme einer HumanTask wird die AnchorNode von dem der Aktion ClaimTask zugeordneten Service im Rahmen der Rechtezuweisung auf die ProcessAggregat-Node des Prozesses gesetzt.

Für die Zuweisung einer Rolle zu einer AnwenderIn werden die Relationen hasRole, hasTempRole, hasInactiveRole, unterschieden.

Zur temporären Rechtevergabe an eine AnwenderIn als

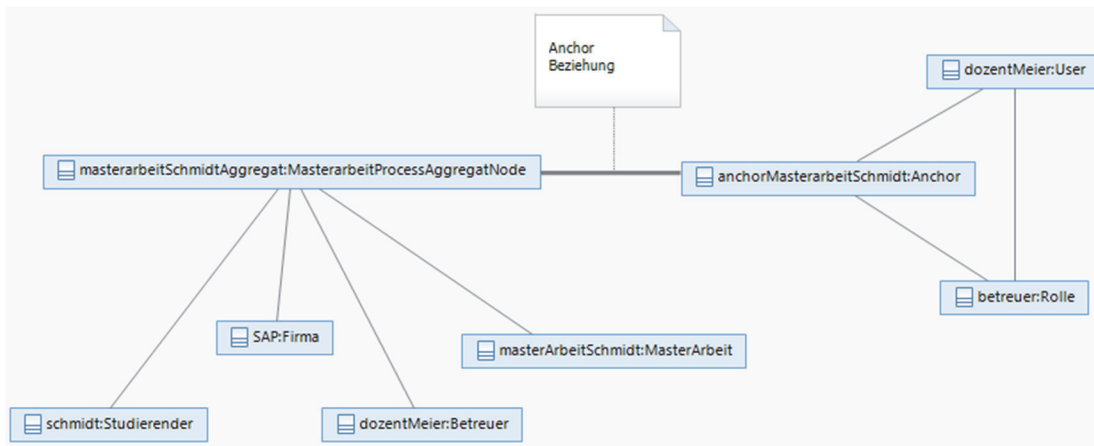


Abbildung 9: AnchorNode-Beispiel

AkteurIn einer HumanTask „hängt“ der der Aktion ClaimTask zugeordnete Service der Stammdatenapplikation die zugeordnete Rolle in die Liste der aktiven temporären Rollen der AnwenderIn.

Dieser Service befüllt ferner die der Rolle und der eingeloggten AnwenderIn zugeordnete Anchor-Node mit der ID der Process Aggregat-Node.

Zum Rechteentzug entfernt der der Aktion TaskDone zugeordnete Service die der HumanTask zugeordnete Rolle wieder aus der Liste der aktiven temporären Rollen und „hängt“ sie in die Liste der nicht aktiven Rollen der eingeloggten AnwenderIn. Die temporäre Rolle wird priorisiert und ersetzt alle anderen Rollen der eingeloggten AnwenderIn.

Die AkteurIn hat während der Erledigung der Task daher nur die der Task zugeordnete temporäre Rolle und ist somit eingeschränkt auf die zur Erledigung der HumanTask benötigten Rechte.

Zerlegung des Prozesses in Use-Case-Szenarien

Der sehr vereinfacht dargestellte Verwaltungsprozess zur Abwicklung einer Masterarbeit beinhaltet die folgenden Use-Case-Szenarien (siehe Abbildung 10 auf der nächsten Seite).

Die StudentIn füllt im Rahmen der Stammdatenapplikation einen Antrag zur Erstellung der Masterarbeit aus und startet damit in der Prozess-Engine den Prozess.

Die Prozess-Engine führt die HumanTask zur Spezifikation der Masterarbeit aus und beauftragt auf diese Weise die BetreuerIn mit der Spezifikation der Masterarbeit. Die BetreuerIn akzeptiert die Aufgabe und spezifiziert unter anderem die Annahme der Betreuung, Thema und Abgabedatum (vgl. Szenario „Claimen einer HumanTask über Repräsentanten“).

Die BetreuerIn quittiert die Erledigung der Aufgabe (vgl. Szenario „Beenden einer HumanTask“).

Die Prozess-Engine holt aus der Datenbank der Stammdatenapplikation die Spezifikation der Masterarbeit und führt das Gateway zur Entscheidung über die Annahme der Betreuung aus (vgl. Szenario „Entscheidungsfindung an einem Gateway“). Die Prozess-Engine sendet eine Mail mit der Spezifikation der Masterarbeit an die

StudentIn. Schließlich schließt die Prozess-Engine den Prozess ab. Einige dieser Szenarien werden in den folgenden Kapiteln ausführlicher erläutert.

Szenario „Claimen einer HumanTask über

„Repräsentanten“

Abbildung 11 (auf der übernächsten Seite) zeigt die Erzeugung eines Repräsentanten der HumanTask der Prozessengine und das darauffolgende Öffnen einer der zugehörigen Webseiten zur Task-Bearbeitung in der Stammdatenapplikation. Zugriffe auf Services sind nicht dargestellt (vgl. Abbildung 2 zur Erläuterung des Kommunikationsdiagramms).

Die Prozess-Engine Camunda bietet die Möglichkeit eigene Ereignis-Listener-Klassen zu erstellen. Dabei handelt es sich um Java-Klassen, die ausgeführt werden, wenn das Token der Prozessinstanz das BPMN-Element erreicht und das konfigurierte Ereignis ausgelöst wird.

Die von der Prozess-Engine aktivierte HumanTask delegiert innerhalb eines solchen Ereignis Listeners die Ausführung der Task an die Stammdatenapplikation.

Zu diesem Zweck erzeugt die Prozess-Engine im Verlauf des Prozesses für jede von einer AkteurIn auszuführende HumanTask über den ProcessAndTaskNotificationService der Stammdatenapplikation eine EngineHumanTask-Node als zugehörigen Repräsentanten innerhalb der Stammdatenapplikation, so z.B. auch für die HumanTask „Masterarbeit Themenausgabe“.

innerhalb der Komponente Prozessverwaltung der Stammdatenapplikation ausgeführt. Die Pflege eines Rechtesystems in der Prozess-Engine entfällt.

Die Liste der EngineHumanTask Nodes wird dabei durch Filterung (Cypher Query) auf diejenigen Nodes reduziert, welche mindestens eine indirekte Beziehung

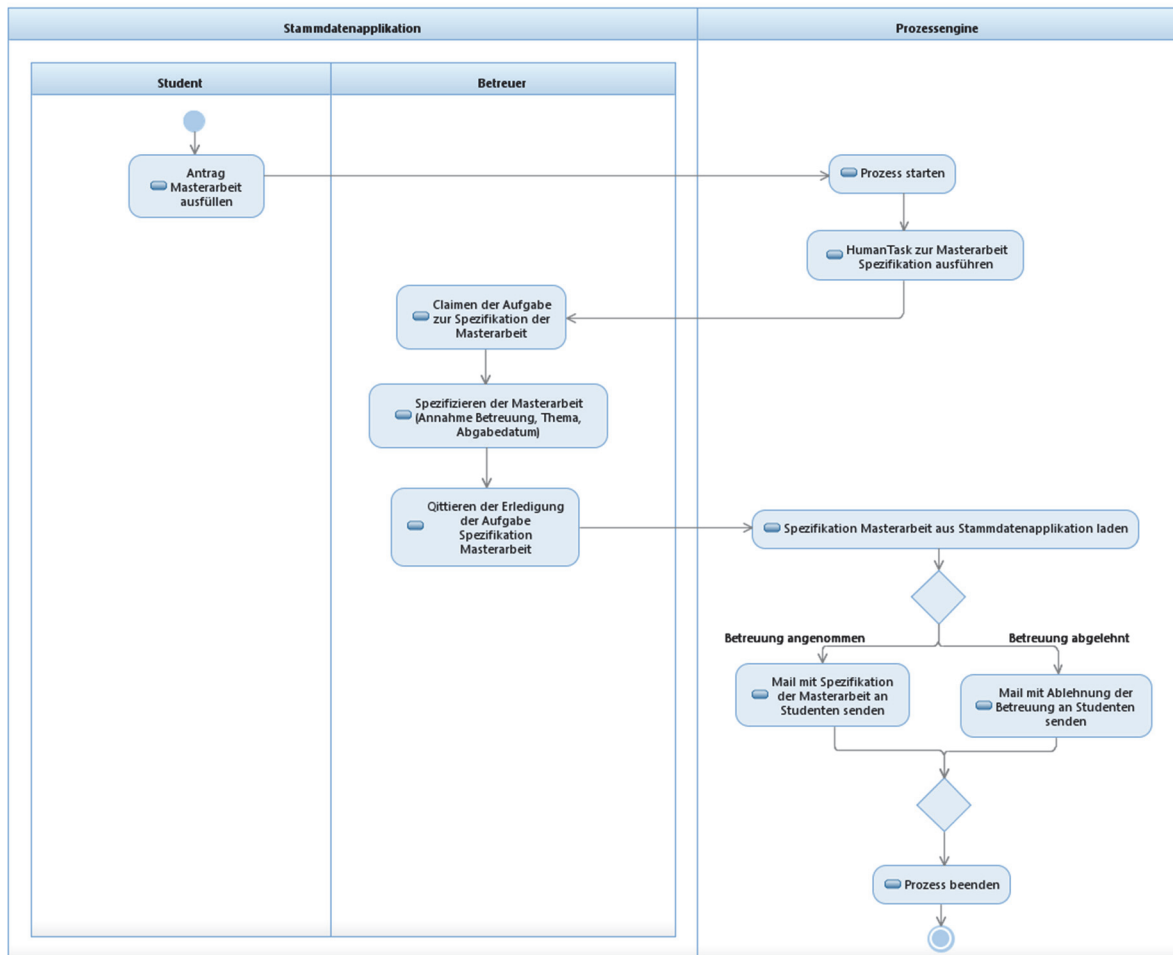


Abbildung 10: Aktivitätsdiagramm Student-Betreuer-Engine

Der aufgerufene ProcessAndTaskNotificationService verbindet diese EngineHumanTask-Node mit derjenigen Node, welche die der HumanTask zugeordnete Rolle repräsentiert (Das Node Label der Rolle entspricht per Konvention dem Namen der HumanTask). Eine entsprechende Zuordnung erfolgt innerhalb der Prozessvariablen von Camunda.

Die innerhalb der Camunda Prozess-Engine vergebene ID der erzeugten Instanz der HumanTask wird dabei als Referenz in ein Attribut der repräsentierenden EngineHumanTask-Node gesetzt.

Schließlich wird die EngineHumanTask-Node als offene Aufgabe der bereits existierenden EngineProzess-Node hinzugefügt.

BenutzerInnen können Aufgaben in Form von EngineHumanTask-Listen und Detail-Seiten einsehen. Das Beanspruchen („Claimen“) und Freigeben einer Aufgabe wird nicht an die Prozess-Engine kommuniziert, sondern

zu einer der inaktiven Rollen der eingeloggtten AnwenderIn haben.

Das „execute“ Recht ermöglicht der AkteurIn das „Claimen“ der selektierten EngineHumanTask.

Durch das „Claimen“ weist die Stammdatenapplikation im Rahmen eines ClaimTaskService der AkteurIn temporär die der auszuführenden HumanTask zugeordnete Rolle zu. Dies geschieht durch „anhängen“ der Rolle in die Liste der aktiven temporären Rollen der AnwenderIn. So erhält die AkteurIn den Zugriff auf alle Entitäten gemäß der durch die Rolle spezifizierten Rechte.

Im Beispiel erhält die betreuende Person der Masterarbeit etwa das Recht, die Nodes StudentIn und Firma zu lesen, die Betreuer-Node zu editieren sowie eine Node MasterArbeit zu erzeugen.

Der Prüfungsausschussvorsitz erhält schließlich die Möglichkeit, die MasterArbeit-Node zu editieren um Fristen und Thema zu prüfen und ggf. ändern zu können.

Darüber hinaus kann der Vorsitz die Nodes StudentIn, Firma und BetreuerIn lesen. Das Lesen der Betreuer-Node ermöglicht etwa die Prüfung der Qualifikation der FirmenbetreuerIn.

Szenario „Entscheidungsfindung an einem Gateway“

Eine AutomaticTask transportiert die im Gateway zur Entscheidung benötigten Daten mittels generischem DataAccessService aus der Stammdatenapplikation in

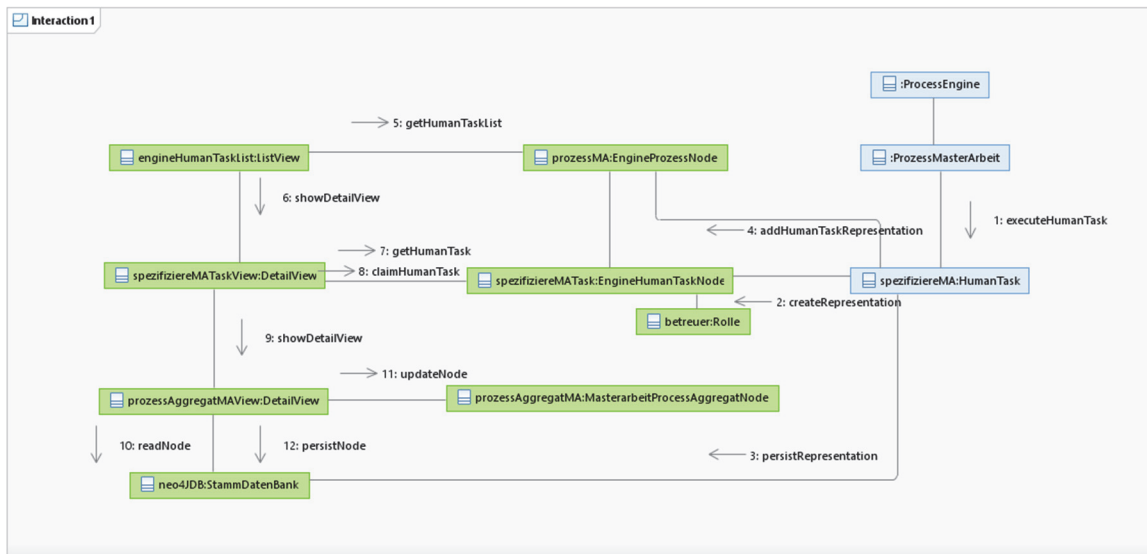


Abbildung 11: Erzeugung eines HumanTask Repräsentanten und „Claimen“

Ggf. könnte man seine Rechte noch dahingehend erweitern, dass er z.B. die von der StudentIn besuchten Wahlpflichtfächer, die Themen vorhergehender Seminare und Praktika einsehen kann.

Aufgrund der Priorisierung der temporären Rollen gegenüber den dauerhaften Rollen wird der Aktionsraum und somit auch die Ansicht der BenutzerIn auf die Bearbeitung der Aufgabe reduziert.

Die BenutzerIn kann die Aufgabe innerhalb des vorgegebenen Aktionsraums, bearbeiten und ist dabei nicht auf feste Reihenfolgen von Ansichten angewiesen.

Szenario „Beenden einer HumanTask“

Wenn die AkteurIn die Erledigung der HumanTask mittels des Buttons TaskDone im ListView der EngineHumanTask-Nodes quittiert, so meldet der zugehörige TaskDoneService die zur Beendigung der HumanTask benötigten Informationen zur Identifikation von Prozess und HumanTask an die Prozess-Engine.

Fachliche Daten, im Beispiel etwa die Spezifikation der Masterarbeit, wurden bereits direkt an die Datenbank der Stammdatenapplikation übermittelt und werden daher nicht an die Camunda-Prozess-Engine übertragen.

Der TaskDoneService entfernt die der HumanTask zugeordnete Rolle wieder aus der Liste der aktiven temporären Rollen und hängt sie in die Liste der nicht aktiven Rollen der eingeloggtten AnwenderIn.

Der ordnungsgemäße Abschluss der HumanTask wird zusammen mit einem Zeitstempel in der EngineHumanTask Node eingetragen und diese aus der Liste der offenen Aufgaben entfernt. Zum Einsehen der Historie wird sie in der Stammdatenapplikation weiter verwaltet.

die Prozessvariablen der Prozess-Engine. Die Spezifikation der benötigten Informationen erfolgt über Cypher Query Strings.

Der von der Prozess-Engine zur Entscheidung benötigte Expression Language Ausdruck wird mittels der Prozessvariablen im Rahmen der Prozessmodellierung formuliert und von der Prozessengine zur Laufzeit ausgewertet.

Im Beispiel ergibt sich etwa aus der Spezifikation der Masterarbeit, dass die BetreuerIn die Arbeit angenommen hat.

FAZIT

Die Prozessmodellierung kämpft mit dem Problem, die Balance zwischen Flexibilität und Strukturiertheit von Prozessabläufen zu finden.

In dieser Arbeit wurde ein imperativer Ansatz als Basis verwendet. Durch die feste mittels BPMN modellierte Ablaufsteuerung können BenutzerInnen eindeutig geführt werden. Sobald die BenutzerIn jedoch eine Aufgabe beansprucht erhält sie durch die zugeordnete Rolle die für eine WissensarbeiterIn wünschenswerte Flexibilität zur Bearbeitung der Aufgabe.

Dies wird erreicht durch eine Reduktion der Prozess-Engine auf die reine Workflow-Steuerung und eine rollenbasierte Delegation der Benutzer-Aufgaben an eine generisch konfigurierbare Stammdatenapplikation.

Aus dem betrachteten Verfahren ergeben sich die nachfolgenden Vorteile:

Erhöhte Flexibilität für WissensarbeiterInnen

Zur Bearbeitung einer HumanTask können der AkteurIn über die zugeordnete Rolle alle Sichten und Funktionen zur Verfügung gestellt werden, welche die Stammdatenapplikation bietet.

Die BenutzerIn kann frei entscheiden, welche Sicht sie wählt oder welche Aktionen sie innerhalb der Bearbeitung ausführt.

Auf diese Weise kann ein Aktivitätsfluss, welcher zur Modellierungszeit nicht vollständig analysiert oder wegen zu hoher Komplexität nicht modelliert wurde, an eine WissensarbeiterIn delegiert werden. Diese kann den Aktivitätsfluss im Rahmen der ihr erteilten Rechte und basierend auf ihrem Prozessverständnis manuell ausführen. Ein an eine WissensarbeiterIn delegierter nicht in der Prozess-Engine modellierter Aktivitätsfluss birgt die Gefahr eines nicht rechtskonformen Ablaufs und kann derzeit nur durch die Einschränkung der Rechte begrenzt werden. Der erhöhte Freiheitsgrad bzw. die Einschränkung der Rechte spiegelt das Vertrauen in die beteiligte WissensarbeiterIn wider. (vgl. auch weiter unten „Bedingte Freischaltung von Webseiten“)

Manuelle Bearbeitung der fachlichen Prozessdaten

Die Stammdatenapplikation ermöglicht eine manuelle nicht von der Prozess-Engine getriggerte Bearbeitung der im Prozess angefallenen Daten. Auf diese Weise können z.B. Korrekturen an den im Rahmen einer HumanTask eingegebenen Daten vorgenommen werden. So kann z.B. das Thema einer Masterarbeit oder das Abgabedatum geändert werden. Ebenso kann ein Prozess für einen nicht modellierten Aktivitätsfluss manuell beendet werden.

Vereinfachtes Verfahren zur Erstellung von automatisierten Prozessen

Prozesse können alleine durch Konfiguration der Entitäten einer Domäne in der Stammdatenapplikation mittels MetaNodes und der Modellierung des BPMN-Workflows in der Prozess-Engine auf hoher Abstraktionsebene ablauffähig spezifiziert werden.

Ein Codieren von Webseiten für HumanTasks entfällt, da die Listen- und Detailsichten zur Datenein- und -ausgabe ebenso wie Sichten zur Vernetzung der Daten auf Basis der spezifizierten MetaNodes erzeugt werden.

Ein aufwändiges Codieren der HumanTask-Prozess-Elemente, z.B. mittels Groovy oder Java, entfällt, da kein Datentransport von den Webseiten in die Prozessvariablen und umgekehrt organisiert werden muss.

Ebenso entfällt ein Transport der fachlichen Daten von Prozessvariablen der Prozess-Engine in die Stammdatenapplikation und umgekehrt.

Daten müssen lediglich für Gateways aus der Stammdatenapplikation in die Prozessvariablen der Engine transportiert werden.

Separation des Workflows vom fachlichen Datenfluss

Das betrachtete Verfahren unterstützt die fachliche Separation des Workflows von den im Rahmen der HumanTasks durchzuführenden Aktionen und ermöglicht damit eine den verschiedenen Abstraktionsebenen angemessene Modellierung und Kapselung in verschiedene Komponenten.

So tangiert z.B. eine Hinzunahme eines Attributes in einer im Rahmen einer HumanTask zu editierenden Entität nicht den modellierten BPMN-Prozess.

Z.B. könnte lediglich durch Editieren der die Entität Masterarbeit spezifizierenden MetaNode ein Attribut Seitenzahl hinzugefügt werden. Der Workflow ist nicht betroffen und die der HumanTask zur Ausgabe des Themas zugeordnete Webseite wird zur Laufzeit aus den Metadaten generiert.

Durch die REST-basierte Kommunikation der Prozess ausführenden Prozess-Engine mit der Stammdatenapplikation können für beide Komponenten unterschiedliche Technologien verwendet werden. Auf diese Weise kann die Workflow Steuerung von der Stammdatenpflege auch technologisch entkoppelt werden.

Für das betrachtete Verfahren ergibt sich aber auch noch Verbesserungspotential bezogen auf nachfolgende Aspekte.

Codeanpassungen der Stammdatenapplikation zur Kopplung an die Prozess-Engine

Die Einbindung einer Prozess-Engine erfordert Codeanpassungen der Stammdatenapplikation z.B. in Form einer Ergänzung der Domäne zur Prozess- und Taskverwaltung sowie der Zuordnung von Prozessen zu Entitäten anderer Domänen. Dies gelingt bei der betrachteten generischen Stammdatenapplikation durch ein Konfigurieren von Entitäten und der Hinzunahme von „SpecialServices“ für das Starten von Prozessen und das „Claimen“ von HumanTasks. Für existierende nicht generische Stammdatenapplikationen ergeben sich an dieser Stelle aufwändige Modifikationen, die nur partiell durch die Wiederverwendung der existierenden Webseiten der Stammdatenapplikation für die Prozessautomatisierung kompensiert werden.

Austauschbarkeit der Prozess-Engine nur eingeschränkt möglich

Das Interface zur Kopplung der Stammdatenapplikation mit der Prozess-Engine bedient die bisher verwendete Camunda-API. Eine andere Prozess-Engine muss ähnliche Konzepte aufweisen, um sich für diese Verwendung zu qualifizieren.

Ein Austausch wird durch die minimale Nutzung des Funktionsumfangs der Camunda-Prozess-Engine für die reine Workflowsteuerung und die Anbindung mittels einfacher REST-Schnittstelle unterstützt.

Erschwert wird ein Austausch im Falle einer Prozess-Engine, die BPMN-Elementen eine andere Semantik zuordnet oder sogar eine andere Prozessbeschreibungssprache (z.B. Petri-Netze) verwendet.

Unvollständiges Errorhandling

Die aktuelle Implementation des evaluierten Beispielprozesses unterstützt keine Rückabwicklung durch Kompensation. Die generische Stammdatenapplikation unterstützt keine Datenbanktransaktionen. Systemreaktionen

für seltene Bedienfehler oder technische Defekte wurden nicht implementiert.

Evaluation komplexer Prozesse

Bisher wurde das System nur anhand eines eher einfachen Prozesses zur Abwicklung einer studentischen Abschlussarbeit getestet.

Dazu wurde im Labor für Softwaretechnologie ein Prototyp im Hinblick auf die beschriebenen wesentlichen Konzepte entwickelt und getestet. Eine weitergehende Evaluation durch Unterstützung mehrerer komplexer Prozesse im Fachbereich ist in Planung.

Bedingte Freischaltung von Webseiten durch Kombination der temporären Rollenvergabe mit einer statusab-hängigen Rollenvergabe

Eine bedingte Freischaltung von Aktionsmöglichkeiten (ähnlich dem CMMN-Wächter-Konzept) könnte in einer zukünftigen Version durch eine statusabhängige Rollenvergabe unterstützt werden.

Pflichten für den Aktor einer HumanTask

Zur Sicherstellung, dass eine „geclaimte“ HumanTask nicht als „done“ beendet wird ohne die eigentliche Aufgabe erfüllt zu haben, könnten der HumanTask im Rahmen der zugehörigen Rolle neben Rechten auch Pflichten zugeordnet werden. Eine solche Pflicht kann z.B. die Erzeugung einer Entität mit Pflichtfeldern sein, z.B. eine Masterarbeit-Node zur Kapselung von Thema und Abgabedatum einer Masterarbeit.

LITERATUR

- Davenport, T.H. 2005. Thinking for a living: how to get better performances and results from knowledge workers. [Buch]. Harvard Business Press, Boston.
- De Giacomo, G. and Dumas, M. and Maggi, F. M. and Montali, M. 2015. "Declarative process modeling in BPMN". In *Proceedings of International Conference on Advanced Information Systems Engineering*. Springer, Cham. 84-100.
- Fahland, D. and Lübke, D. and Mendling, J. and Reijers, H. and Weber, B. and Weidlich, M. and Zugal, S. 2009. "Declarative versus imperative process modeling languages: The issue of understandability". In *Enterprise, Business-Process and Information Systems Modeling*. Springer, Berlin, 353-366.
- Hinkelmann, K. and Pierfranceschi, A. 2014. "Combining process modelling and case modelling". In *Proceedings of 8th International Conference on Methodologies, Technologies and Tools enabling e-Government MeTTeG14*, 83.
- Hollingsworth, D., and Hampshire, U. K. 1995. *Workflow management coalition: The workflow reference model*. Document Number TC00-1003, 19, No. 16, 224.
- OMG. 2013. *Business Process Model and Notation (BPMN), Version 2.0.2*.
- OMG. 2016. *Case Management Model and Notation (CMMN) Version 1.1*.
- Pesic, M. 2008. *Constraint-based workflow management systems: shifting control to users*. Technische Universität Eindhoven.
- Reijers, H. A. and Rigter, J. H. M. and van der Aalst, W. M. 2003. The case handling case. *International Journal of Cooperative Information Systems*, 12, No. 3, 365-391.
- Ter Hofstede, A. H. and Van der Aalst, W. M. and Adams, M. and Russell, N. 2009. *Modern Business Process Automation: YAWL and its support environment*. Springer Science & Business Media.
- van Der Aalst, W. M. and Pesic, M. and Schonenberg, H. 2009. "Declarative workflows: Balancing between flexibility and support". *Computer Science-Research and Development*, 23, No. 2, 99-113.

KONTAKT

Prof. Dr. Peter Edelmann
Technische Hochschule Mittelhessen
Wilhelm-Leuschner-Straße 13
61169 Friedberg
Peter.Edelman@mnd.thm.de

Prof. Dr. Armin Wagenknecht
Technische Hochschule Mittelhessen
Wiesenstraße 14
35396 Gießen
Armin.Wagenknecht@mni.thm.de

M.Sc. Michel Jonathan Schmitz
Technische Hochschule Mittelhessen
Wiesenstraße 14
35396 Gießen
michel.jonathan.schmitz@its.thm.de