

Studienarbeit

simBPMN Visualisierungseditor Analyse

Visualisierungssoftware

Herbstsemester
2021

Autoren:	Michel Mirsayyah Sven Höpfner
Betreut von:	Prof. Dr. Andreas Rinkel
Abgabe:	24.12.2021
Version:	1.0

Inhaltsverzeichnis

1	Aufgabenstellung	1
1.1	Einleitung	1
1.2	Auftrag	1
1.2.1	Plugin	1
1.2.2	Editor Erweiterung	1
1.2.3	Neuentwicklung	1
2	Technischer Bericht	3
2.1	Einleitung	3
2.1.1	Ausgangslage & Problemstellung	3
2.1.2	Ziele	3
2.2	Beschreibung BPMN	4
2.2.1	Events	5
2.2.2	Gateways	5
2.2.3	Beispiel	6
2.3	Beschreibung simBPMN	7
2.3.1	Architektur/Struktur (Entity-Flow)	7
2.3.2	Prozesslogik (Token-Flow)	7
2.4	Vision für simBPMN	8
2.4.1	Benutzeroberfläche	8
2.4.2	Beschreibungsmöglichkeit von simBPMN	8
2.5	Evaluation von bestehender Software	9
2.5.1	Cross-Plattform Enabler	9
2.5.2	Cross Platform Frameworks	10
2.5.3	Frameworks	13
2.5.4	Engines	15
2.5.5	Anwendersoftware	22
2.5.6	Weitere Tools	25
2.5.7	Ausgearbeitete Variationen	26
2.5.8	Fazit	27
3	Projektplan	29
3.1	Projektorganisation	29
3.1.1	Organisationsstruktur	29
3.1.2	Externe Schnittstellen	29
3.2	Managementabläufe	30
3.2.1	Zeitbudget	30
3.2.2	Zeitliche Planung	30
3.2.3	Besprechungen	32
3.3	Risikomanagement	32
3.3.1	Risikoanalyse	33
3.3.2	Umgang mit Risiken	34
3.4	Arbeitspakete	34
3.5	Infrastruktur	34
3.6	Qualitätsmassnahmen	35

3.6.1	Code Reviews	35
3.6.2	Guidelines	35
4	Anforderungsspezifikationen	37
4.1	Produktfunktionen	37
4.2	Lieferumfang	37
4.3	Benutzer Charakteristik	37
4.4	Annahmen, Einschränkungen und Abhängigkeiten	38
4.4.1	Annahmen	38
4.4.2	Einschränkungen	38
4.4.3	Abhängigkeiten	38
4.5	Use Cases	39
4.5.1	Use Case Diagramm	39
4.5.2	Beschreibungen (Fully Dressed)	40
4.5.3	Beschreibungen (Brief)	44
4.6	Nicht-funktionale Anforderungen	45
4.6.1	Funktionalität	45
4.6.2	Zuverlässigkeit	45
4.6.3	Effizienz	45
4.6.4	Benutzbarkeit	45
4.6.5	Sicherheit	45
4.6.6	Wartbarkeit	46
4.6.7	Übertragbarkeit	46
5	Schlussbericht	47
5.1	Zielerreichung	47
5.2	Review der eingesetzten Tools	47
5.3	Persönliche Erfahrungen	48
5.3.1	Michel Mirsayyah	48
5.3.2	Sven Höpfner	49
6	Zeitauswertung	51
6.1	Zeitauswertung pro Person	51
6.2	Zeitauswertung pro Person und Woche	52
7	Kompendium	53
7.1	KNIME	53
7.1.1	KNIME Nodes erstellen	53
7.1.2	KNIME Workflow Editor	54
7.2	Camunda Modeler	55
7.2.1	Properties Panel Hinzufügen	55
7.2.2	Palette anpassen	55
8	Referenzen	59
9	Eigenständigkeitserklärung	60
10	Glossar	61
A	Sitzungsprotokolle	63
A.1	Kick-Off Meeting	63
A.2	SimBPMN Workshop Meeting	64
A.3	Meeting zum Meilenstein 1 (Projektplan)	65
A.4	Vorstellungsmeeeting KNIME	66
A.5	Knime und simBPMN	67
A.6	Vorstellungsmeeeting KNIME	68
A.7	Vorstellungsmeeeting Alternative	69

A.8	Vorstellungsmeeeting Camunda Modeler & BPMN Studio	70
A.9	Fortschritt Camunda Modeler	71
A.10	Vorbereitung Zwischenpräsentation	72
A.11	Feedback Zwischenpräsentation	73
A.12	Abschluss Entwicklung	74
A.13	Abschluss Studienarbeit	75

Bildverzeichnis

2.1	Einfacher Prozess in BPMN	6
2.2	simBPMN Beispiel: Formung von Blechen	8
2.3	Avalonia UI	11
2.4	Uno UI	12
2.5	BPMN RPA Studio	13
2.6	Core2D UI	16
2.7	Workflowengine UI	17
2.8	Camunda Modeler UI	18
2.9	Viewflow BPMN Diagram	21
2.10	BPMN IO UI	22
2.11	BPMN IO UI	23
2.12	Camunda Modeler UI	24
2.13	BPMN View UI	25
3.1	simBPMN Zeitplan	30
4.1	Use Case Diagramm	39
6.1	Zeitauswertung pro Person	51
6.2	Zeitauswertung pro Person und Woche	52
7.1	Javascript mit base64 SVG Bild	56
7.2	Index.js	58

Tabellenverzeichnis

2.1	BPMN Standardelemente	4
2.2	BPMN Events	5
2.3	BPMN Gateways	5
2.4	simBPMN Elemente	7
2.5	Cross Platform Frameworks	10
2.6	Evaluation tabellarische Übersicht	27
3.1	Projektorganisation und Verantwortung	29
3.2	Externe Personenschnittstellen	29
3.3	Meilensteine	31
3.4	Risikomanagement	33
3.5	Qualitätsmassnahmen	35
3.6	Branches	36
4.1	Use Case: Prozess Modellieren	40
4.2	Use Case: Nested Prozesse	41
4.3	Use Case: Remarks und Files hinterlegen	42
4.4	Use Case: Remarks und Files hinterlegen	43

Aufgabenstellung

1.1 Einleitung

In der Geschäftswelt existieren viele unterschiedliche Beschreibungsformen um einen Prozess darzustellen, dabei bringen alle Notationsformen ihre eigenen Vor- und Nachteile mit sich. Um nun die Vorzüge dieser Formen zu vereinen wurde die Business Process Model and Notation, kurz BPMN, geschaffen. Mit BPMN hat die [Object Management Group](#) eine Notation eingeführt, welche als eine Art Standard angesehen werden kann. Man kann diese als [UML](#) der Geschäftsprozesse ansehen. Mit der [BPMN 2.0](#) Erweiterung kamen zu bestehenden Symbolen noch Events dazu. Damit ist man nun im Stande komplexe Prozesse mit zeitlichen Bedingungen abzubilden.

BPMN eignet sich im Kern ebenfalls dazu Simulationsprozesse zu beschreiben. Zusammen mit Doktoranden der Universität der Bundeswehr in München hat Prof. Dr. Andreas Rinkel die [simBPMN](#) Notation ausgearbeitet, welche es ermöglicht Prozesse mit Simulationsbestand mittels einer abgewandelten Version von BPMN zu beschreiben.

1.2 Auftrag

Der Sinn dieses Vorhabens ist es zu überprüfen ob die simBPMN-Notation, angewendet in einem Programm, praxistauglich ist. Im Rahmen der Studienarbeit soll eine Vorarbeit geleistet werden, welche zu einem späteren Zeitpunkt durch eine Folgearbeit umgesetzt wird. Die Vorarbeit beinhaltet eine tiefgehende Evaluation bestehender Software und dessen Tauglichkeit für simBPMN, dabei soll allerdings aber eine Neuentwicklung nicht ausgeschlossen werden. Mit dem besten Tool, gemäss der Evaluation, soll ein technologischer Durchstich gemacht und festgehalten werden wie mit diesem aus Sicht des Programmierers interagiert werden muss. Der in der nachfolgenden Arbeit erstellte Prototyp muss damit simulations-gestützte Prozesse mittels simBPMN sauber beschreiben können. Folgende Ansätze sollen in der Evaluation berücksichtigt werden.

1.2.1 Plugin

Bei einem Plugin geht es darum die Funktionalität eines Editors zu erweitern. Dies umfasst es neue [Nodes](#) einzuführen, welche der simBPMN Notation entsprechen und diese in eventuelle Funktionen mit einzubinden. Diese Funktionen können bereits existieren oder neu entwickelt werden.

1.2.2 Editor Erweiterung

Sollte es nicht möglich sein in einem bestehendem Editor ein Plugin einzufügen, weil dieses zum Beispiel nicht den Anforderungen gerecht wird, dann wäre die nächste Option einen bereits bestehenden Editor programmatisch zu erweitern. Dabei muss geachtet werden, dass der Code von dem Editor einem zur Verfügung steht und man diesen nach den Lizenzbedingungen auch benutzen darf.

1.2.3 Neuentwicklung

Sollte es nicht möglich sein ein Plugin oder eine Erweiterung zu erstellen oder dies mit extremen Aufwand verbunden sein, kann eine Neuentwicklung in Betracht gezogen werden.

Technischer Bericht

2.1 Einleitung

2.1.1 Ausgangslage & Problemstellung

Um Businessprozesse zu beschreiben existiert derzeit die [BPMN 2.0](#) Notation. Mit dieser können bereits sehr viele Prozesse dargestellt werden. Allerdings bietet BPMN so viele Elemente an, dass die pure Auswahl teilweise den Anwender überfordern könnten, ausserdem begrenzt sich BPMN auf die Prozesslogik. Ein Aspekt auf die Architektur des Prozesses fehlt vollkommen.

Mithilfe von [simBPMN](#) werden die Anzahl Elemente von BPMN auf ein Minimum beschränkt. Das Konzept des Entity-Flows dabei neu eingefügt werden, da dies allerdings ein komplett neues Konzept ist, besteht noch keinerlei Unterstützung, wie beispielsweise ein Editor in welchem diese Notation dargestellt wird. Im besten Fall soll eine bereits bestehende Software um diesen Teil erweitert werden.

2.1.2 Ziele

Das Ziel von simBPMN ist eine Systembeschreibung und Modellierungsunterstützung auf höchster Abstraktionsstufe zu ermöglichen. Ein Benutzer soll möglichst unkompliziert ein System beschreiben können. Um die Erlernbarkeit und Verständlichkeit zu unterstützen, gilt es die Anzahl der Modellierungskomponenten beziehungsweise Symbole so gering wie möglich zu halten. Mithilfe eines Editors kann ein Prozess in der simBPMN Notation beschrieben werden. Die Elemente des Prozesses können im Editor angepasst werden und es ist möglich ihnen weitere Attribute zuzufügen, um diese mit allfälligen Informationen wie Daten oder angehängten Files tiefer zu beschreiben.

Es soll die Möglichkeit geben den Editor in folgenden Arbeiten erweitern zu können um dort eventuelle neue Funktionalitäten mit einzubauen.

2.2 Beschreibung BPMN

Die Business Process Model and Notation Spezifikation, kurz [BPMN 2.0](#), bietet eine grafische Notation um Geschäftsprozesse in einem Diagramm abzubilden. Das Ziel dieser Notation ist es eine Darstellung zu bieten, welche einfach zu verstehen für Business-Leute und eine komplexere Prozessemantik beinhaltet für technisch-versierte Benutzer.


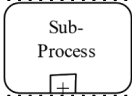



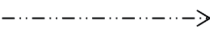
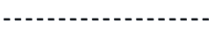





Element	Name	Beschreibung
	Task	Ein Task ist ein Bestandteil eines Prozesses und kann nicht weiter unterteilt werden, ist also Terminal
	Subprozess	Ein Subprozess enthält einen oder mehrere Tasks oder Subprozesse
	Event	Ein Event beschreibt etwas, was zu einem bestimmten Zeitpunkt im Prozess passiert und kann dabei Tokens erzeugen
	Gateway	Mit Gateways werden Fallunterscheidungen dargestellt
	Sequence Flow	Durch den Sequence Flow kann man die Richtung des Geschäftsprozesses definieren
	Message / Material Flow	Eingehende Nachrichten / Materialien in den Geschäftsprozesses können damit dargestellt werden
	Association	Durch eine Association kann man den Zusammenhang von Objekten erkennen
	Data Object	Ein Data Object repräsentiert ein Artefakt, welches der Geschäftsprozess bearbeitet
	Text Annotation	Eine Annotation ist ein Kommentar, der einem Element eines Geschäftsprozesses zugeordnet werden kann
	Group	Eine Group ist ein Hilfsmittel, um Elemente eines Geschäftsprozesses visuell zusammenzufassen
	Pool	Der Pool ist ein Hilfsmittel, um einen Elementfluss darzustellen
	Lanes	Die Lanes können verwendet werden um Prozessübergreifende Geschäftsprozesse darzustellen

Tabelle 2.1: BPMN Standardelemente

2.2.1 Events

Events können entweder geworfen oder gefangen werden. Das geworfene Event wird an alle eingetragenen Catcher weitergegeben. Diese können das Event selbstständig abarbeiten oder verwerfen.

Die Events und Gateways werden dann nochmal in die folgenden Kategorien unterteilt:




Element	Name	Beschreibung
	Start-Event	Beschreibt den Anfang eines Prozesses
	Intermediate-Event	Ein Intermediate-Event kann während des Prozesses jederzeit geworfen und gefangen werden
	End-Event	Beschreibt das Ende eines Prozesses

Tabelle 2.2: BPMN Events

2.2.2 Gateways

Gateways beschreiben Verzweigungen im Prozessverlauf.

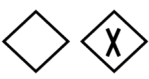



Element	Name	Beschreibung
	Data-based Exclusive Gateway	Durch das Data-based Gateway können einfache Fallunterscheidungen gehandhabt werden. Diese können datenbasiert, zufällig oder an Bedingungen geknüpft sein. Des weiteren können diese Gateways zum Zusammenführen der Datenströme genutzt werden
	Parallel Gateway	Durch das Parallel Gateway kann ein Datenstrom in zwei Parallele Datenströme aufgeteilt werden
	Inclusive Gateway	Durch das Inclusive Gateway wird ein Datenstrom in eine, mehrere oder alle Möglichkeiten aufgeteilt. Dies ist abhängig davon, welche Parameter der Token besitzt
	Event-based Exclusive Gateway	Das Event-based Gateway wartet auf ein Event und führt je nach dem, welches Event gefangen wird einen anderen Vorgang aus

Tabelle 2.3: BPMN Gateways

2.2.3 Beispiel

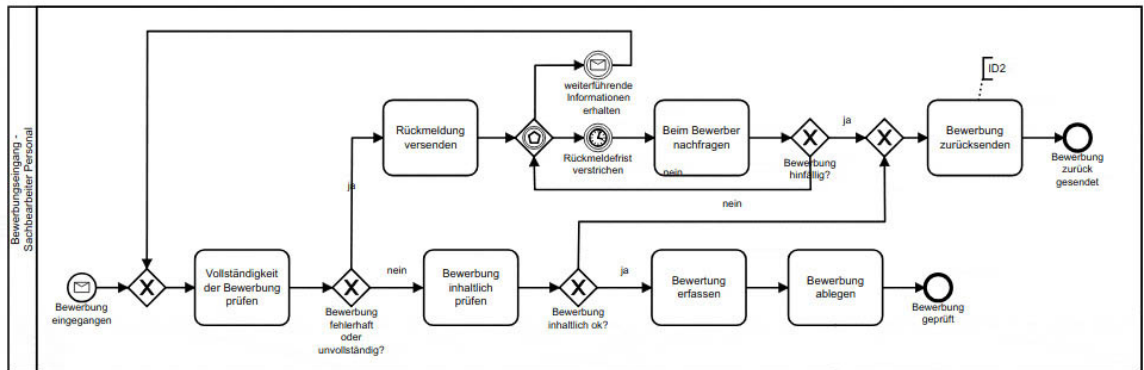
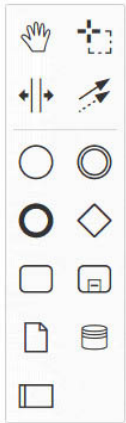


Bild 2.1: Einfacher Prozess in BPMN

2.3 Beschreibung simBPMN

simBPMN ist eine Abwandlung des BPMN 2.0 Standards.

In simBPMN werden einige der Standard Diagrammelemente von BPMN verwendet um somit Geschäftsprozesse zu beschreiben. Dies dient der Vereinfachung der Notation, damit die Benutzer nicht so schnell überfordert werden und die Diagramme auf einem simplen Level bleiben. simBPMN erweitert die Notation von BPMN auch noch um ein paar eigene Elemente um den Benutzern die Möglichkeit zu geben die Architektur eines Prozesses zu beschreiben.

Zu diesen BPMN-Elementen kommen nun noch folgende simBPMN Komponenten hinzu.

Element	Name	Beschreibung
1	Prozess	Ein Prozess führt ein beschriebenes Verhalten aus
2	Entity	Eine Entity läuft mithilfe von Sequence-Flows von Prozess zu Prozess und löst bei der Ankunft das Start-Event aus wodurch ein Token erstellt wird.
3	Prozess-Step	Ein Prozess-Step wird von einem Token abgearbeitet
4	Token	Ein Token fließt durch einen Prozess und besitzt dabei keinen Zustand. Allerdings kann er die Zustände von Entities, Ressourcen oder andere Prozesse ändern, sowie Events werfen
5	Resources	Wird für die Ausführung eines Prozesses benötigt

Tabelle 2.4: simBPMN Elemente

2.3.1 Architektur/Struktur (Entity-Flow)

Entities bestimmen den Ablauf des Hauptprozesses und werden dabei als Input für die Businessprozesse verwendet. Die Resultate der Businessprozesse erstellen dann als Output neue Entities, welche weitergegeben werden.

2.3.2 Prozesslogik (Token-Flow)

Ein Token ist ein Status. Dieser durchläuft einen Businessprozess und ändert dabei seinen Status. Durch den Token kann bestimmt werden in welche Richtung sich der Prozess bewegt.

2.4 Vision für simBPMN

Weiter oben wurde bereits erläutert wie und mit welchen Elementen **BPMN 2.0** Prozesse modelliert werden können. Die Idee von **simBPMN** ist es ebenfalls BPMN zu nutzen um den Simulationsanteil eines Elementes zu beschreiben. Dabei soll zu jedem Element ein default Prozess abgebildet werden können. Dieser default Prozess kann anschliessend bei einzelnen Elementen wieder individuell überschrieben werden. Technisch ausgedrückt erbt jedes Element eine Basisfunktionalität welche angepasst werden kann. So ist es möglich mit relativ wenig Aufwand ein Modell zu beschreiben.

2.4.1 Benutzeroberfläche

Um den Modellierungsprozess weiter zu vereinfachen sollen es dem Benutzer möglich sein eigene definierte Felder, beispielsweise Bemerkungen, auf Elementen zu machen. Weiter können Dokumente zu Elementen hinterlegt werden welche einen Prozess oder das Element selbst genauer definieren. Einfachheit wird hier gross geschrieben, deshalb sollen auf den ersten Blick auch nur die wichtigsten Informationen in der Benutzeroberfläche dargestellt werden. Der Benutzer soll hier möglichst ohne aufwand ein Modell abbilden können und dazu die bestmögliche Unterstützung vom Tool erhalten.

2.4.2 Beschreibungsmöglichkeit von simBPMN

Um detaillierter aufzuzeigen wie die Simulationsbeschreibung von **simBPMN** aussieht, wird im folgenden ein kleines Beispiel gemacht. Wir nehmen an wir beschreiben eine Fabrik welche Bleche stanzt und in Form biegt. Nach dem formen werden die Bleche geprüft und anschliessend gelagert. Falls ein Blech nicht sauber geformt wurde, wird dieses erneut geformt und geprüft. Dies könnte wie folgt aussehen:

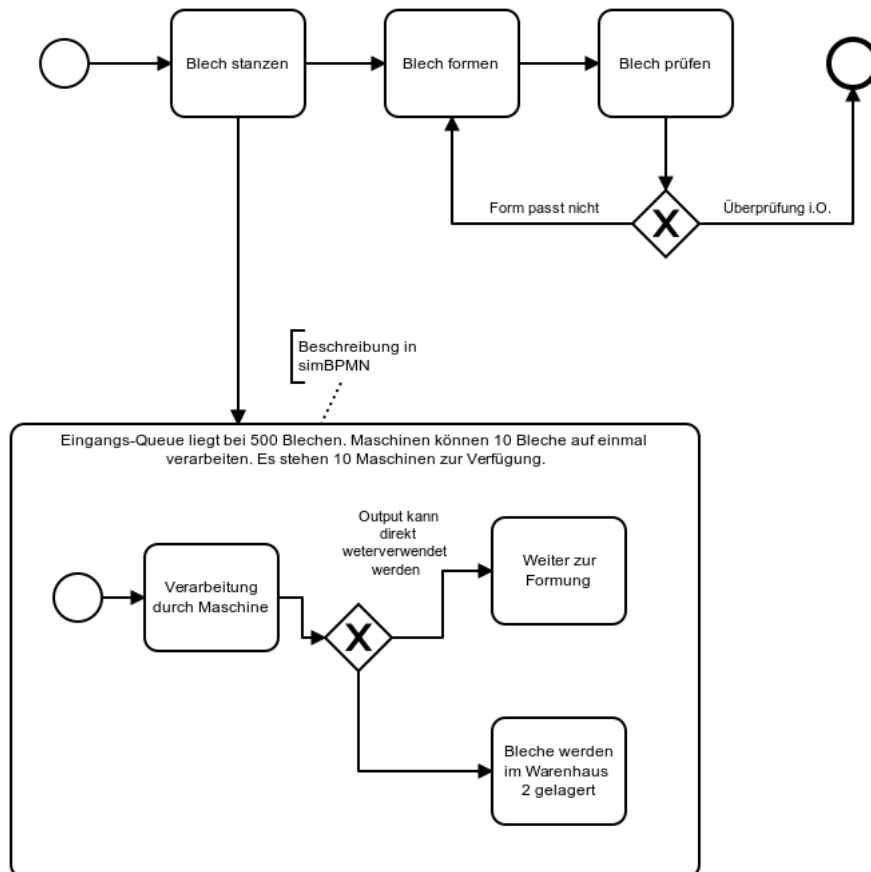


Bild 2.2: simBPMN Beispiel: Formung von Blechen

Hier wird ebenfalls mit BPMN beschrieben wie sich der Prozess innerhalb von "Blech stanzen" verhält. Man kann erkennen wie gehandelt werden muss falls beispielsweise nicht die komplette verarbeitete Menge an Blechen direkt an die Formungsmaschine weitergegeben werden kann.

2.5 Evaluation von bestehender Software

Ziel der Evaluation war es bestehende Software zu finden, welche unseren Use-Cases teilweise oder ganz abdecken. Diese sollen anschliessend als Basis weiterverwendet werden, somit mussten wir nicht das Rad neu erfinden. Es gibt eine unglaubliche Anzahl an [BPMN 2.0](#) Tools die verfügbar sind, leider sind aber meistens nur die kaufbaren brauchbar. Für die Studienarbeit wollten wir aber ein Tool erstellen, welches auch später öffentlich zugänglich und erweiterbar ist, dafür mussten wir uns auf Open-Source Projekte fixieren. Glücklicherweise findet man im Internet Webseiten welche solche Open-Source Auflistungen[1] für BPMN machen. An dieser Stelle muss noch erwähnt werden dass natürlich nicht nur Tools von der angegebenen Website analysiert, sondern auch eigenständig recherchiert wurde.

2.5.1 Cross-Plattform Enabler

Cross-Plattform Enabler sind Frameworks welche eingesetzt werden können um User-Interfaces Plattformunabhängig zu erstellen. Diese besitzen weder die Fähigkeit ein BPMN Diagramm zu erstellen noch dieses auszuführen. Sie sind lediglich dazu da einen höheren Abstraktionslevel aus Programmiersicht zu erzeugen.

2.5.2 Cross Platform Frameworks

Sollte sich keine Applikation finden, welche einfach verwendet oder erweitert werden kann, muss diese wahrscheinlich von Hand entwickelt werden. Dafür wurden die bekanntesten Cross Platform Frameworks analysiert und mithilfe von Pros und Contras bewertet. Diese Analyse bietet ausserdem den Vorteil, dass man herauslesen kann womit wahrscheinlich die meisten bisherigen Applikationen entwickelt wurden. Da es eine Mehrheit bei den Webtechnologien wie JavaScript bei diese Frameworks gibt, wird man wahrscheinlich in diesem Bereich die vielversprechendsten Applikationen antreffen

Name	Sprache	Pros	Contras
Xamarin	C#	<ul style="list-style-type: none"> • Viel Erfahrung C# im Team • Weit verbreitetes Framework • All Use-Cases können damit umgesetzt werden • Es können bereits bestehende BPMN Nuget Packages verwendet werden 	<ul style="list-style-type: none"> • Schlechte Erfahrungsberichte von Personen, welche das Framework bereits eingesetzt haben
Flutter	Dart	<ul style="list-style-type: none"> • Erfahrung mit Flutter im Team 	<ul style="list-style-type: none"> • Desktop Apps sind noch in der Beta • Kein definitiver Support für Drag and Drop.
Ionic	HTML, CSS, and JavaScript	<ul style="list-style-type: none"> • JavaScript bietet Libraries für BPMN • Gute Erfahrungsberichten von Personen 	<ul style="list-style-type: none"> • Keine Erfahrung mit Javascript im Team • Kann bei Komplexen Systemen zu Performance Problemen kommen
React Native	JSX (Extended JavaScript)	<ul style="list-style-type: none"> • Besser Performance als bei Alternativen 	<ul style="list-style-type: none"> • Probleme bei Upgrades • Mit Support von Libraries wird teilweise einfach aufgehört • Keine Erfahrung mit JSX im Team
Electron	HTML, CSS, and JavaScript	<ul style="list-style-type: none"> • Im Gegensatz zu normalen Webapps können Files verändert und geschrieben werden 	<ul style="list-style-type: none"> • Die Applikation benötigt viel Speicherplatz (mind. 100mb) • Keine Erfahrung mit Javascript im Team

Tabelle 2.5: Cross Platform Frameworks

C# - Avalonia

Beschreibung Avalonia ist eine Open-Source Lösung als alternative für [WPF](#). Die zu Verfügung gestellten Komponenten sind Plattformunabhängig und können einfach eingebunden werden.

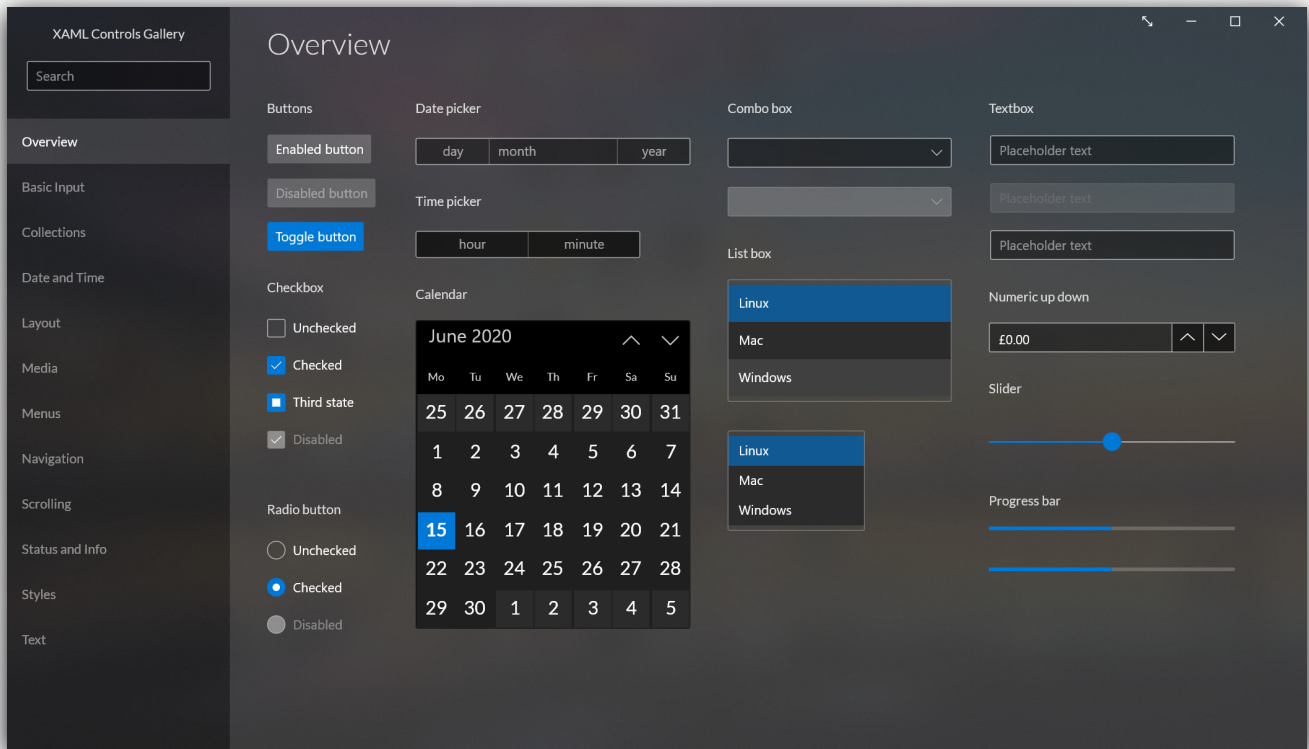


Bild 2.3: Avalonia UI

Screenshots

Pro

- Plattformunabhängig
- Bietet WPF-ähnliche Controls

Contra

- Keine BPMN Funktionalität geboten
- Nicht sauber dokumentiert
- Alles ausserhalb von Base Use Cases ist nicht implementierbar (Bsp. custom events)

Referenzen

- [Avalonia](#)
- [Avalonia GitHub](#)

C# - Uno Platform

Beschreibung Uno ist ein Framework zur Implementation von Plattformunabhängigen Anwendungen. Diese ermöglicht es auf fast allen Plattformen inklusive Android und IOS UIs zu erstellen. Uno verspricht dabei Pixel Perfect Rendering und native performance.

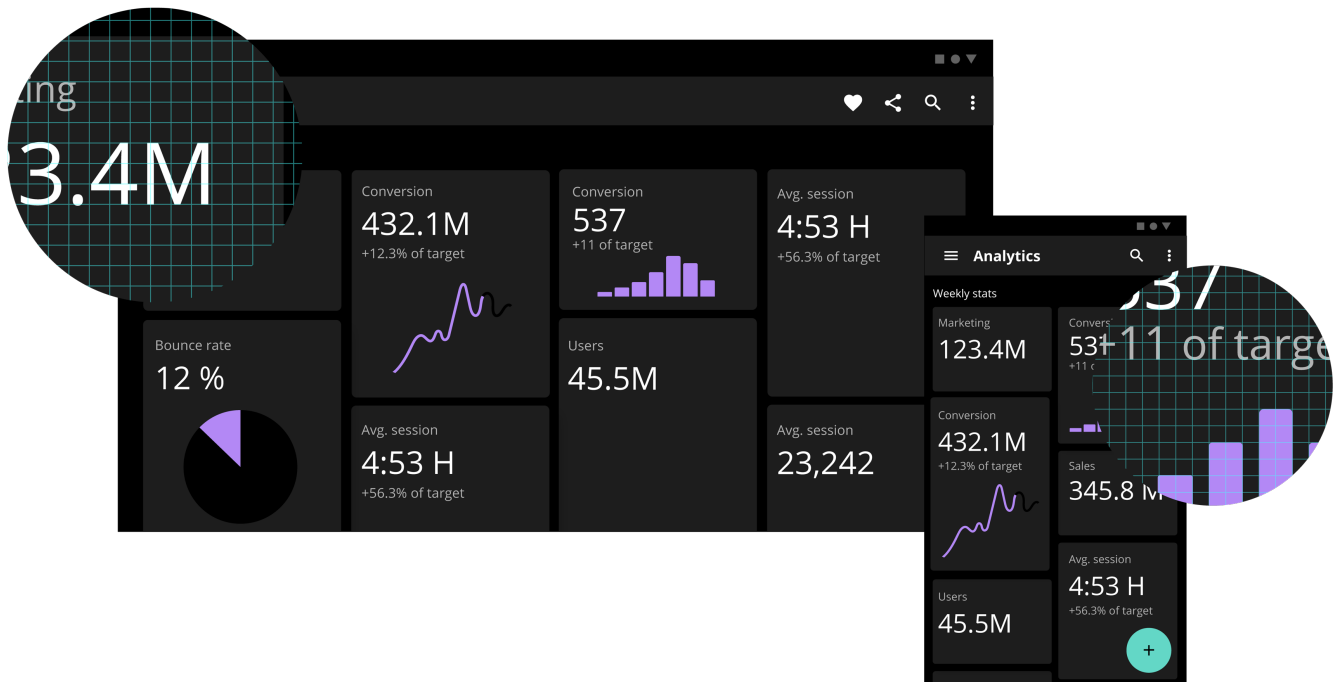


Bild 2.4: Uno UI

Screenshots

Pro

- Plattformunabhängig
- Bietet in etwa die Funktionalität von .Net Maui

Contra

- Keine BPMN Funktionalität geboten
- UIs müssen pro Plattform definiert werden

Referenzen

- [Uno Platform](#)
- [Uno GitHub](#)

2.5.3 Frameworks

Die hier vorgestellten Anwendungen funktionieren nicht eigenständig als Komplettlösung und sollten als Komponenten angesehen werden, welche BPMN verarbeiten können. Die Idee ist es eine Workflow Engine einzusetzen mit welcher man Diagramme zeichnen kann und ein Framework um dieses auszuführen bzw. zu verifizieren.

Python - BPMN RPA

Beschreibung BPMN RPA ist ein Framework geschrieben in Python welches ermöglicht BPMN Diagramme abzubilden. Dieses alleine reicht aber nicht um Diagramme zu erzeugen. Um Flows abzubilden müssen weitere Programme eingesetzt werden: "There is no need for installing BPMN-RPA Studio, DrawIO or Visio to run the flows. Installing one of these applications is only needed for creating the flows.". Das hier angesprochene BPMN RPA Studio würde super für unsere Anwendungszwecke passen, ist jedoch closed source.

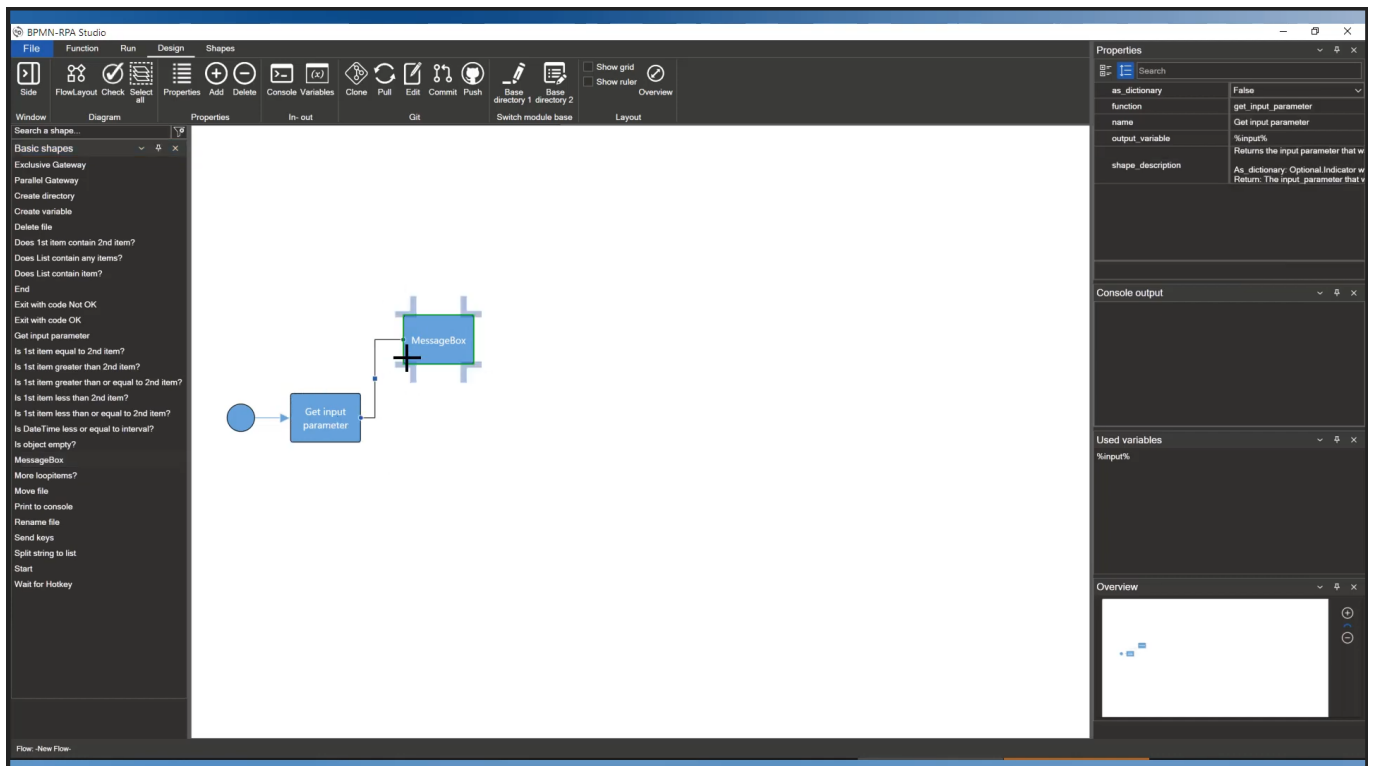


Bild 2.5: BPMN RPA Studio

Screenshots

Pro

- Plattformunabhängig
- BPMN Funktionalität geboten

Contra

- BPMN RPA Studio ist closed source
- Andere Programme sind nur "Zeichnungs-Tools" und bringen keine Funktionalität mit sich

Referenzen

- [BPMN RPA GitHub](#)
- [BPMN RPA Studio Video](#)

Python - Spiffworkflow

Beschreibung Spiffworkflow ist eine weitere Workflow-Engine geschrieben in Python. Diese ermöglicht es BPMN Diagramme zu parsen und auszuführen.

Pro

- Bietet BPMN Support an
- Plattformunabhängig

Contra

- Kein UI

Referenzen

- [Spiffworkflow GitHub](#)

2.5.4 Engines

Engines werden hier als Anwendungen bezeichnet welche es ermöglichen einen Prozess als Diagramm darzustellen. Mit Prozessen ist nicht BPMN gemeint sondern allgemeine Prozesse. Eine Engine kann mittels eines Frameworks BPMN fähig gemacht werden.

KNIME

Beschreibung [KNIME](#) ist eine auf [Eclipse](#) basierende OpenSource Software womit Prozesse erstellt werden können. In der Applikation ist es möglich die Ausführung des Prozesses direkt in die Beschreibung mit einzubauen.

Potential [KNIME](#) bietet bereits sehr viel Funktionalität um Prozesse darzustellen und mit diesen zu arbeiten. Man kann einfach neue Nodes erstellen und diese verbinden. Es besteht die Möglichkeit sogenannte Metanodes zu erstellen, mit welchen man Subprozesse darstellen kann.

KNIME Analytics Platform Funktionalitäten Mithilfe der [KNIME Analytics Platform](#) könne Prozesse definiert werden. Die Platform kann auch mithilfe einer Extension, welche mit Java entwickelt wird, um verschiedene Nodes erweitert werden.

Nodes in [KNIME](#) beschreiben jeweils eine Aufgabe und können mit anderen Nodes verbunden werden. Der In- und Output Typ von Nodes ist konfigurierbar. Des weiteren besteht die Möglichkeit die Nodes so zu konfigurieren, dass gewisse Daten vom User gesetzt werden müssen, welche von [KNIME](#) validiert werden.

Metanodes ermöglichen es mehrere Nodes zusammenzufassen. Dadurch können Subprozesse von [simBPMN](#) definiert werden. Ein Metanode besitzt wie ein normales Node auch In- und Outputs, welche im Node-Prozess selber angesprochen werden können.

Workflows welche mit [KNIME](#) erstellt wurden können ausserdem auch direkt ausgeführt werden. Sollten die Nodes also konfiguriert sein und entsprechende Funktionalität enthalten, können Daten mithilfe des Workflows manipuliert und ausgewertet werden.

Extensions in [KNIME](#) können mithilfe des [KNIME-SDK](#) [2] erstellt werden. Die Extension wird in [Java](#) geschrieben und kann relative einfach in die [KNIME Analytics Platform](#) eingebunden werden. Der Setup einer Extension ist im Internet sehr gut beschrieben, allerdings findet man zu anderen Problemen keine Hilfestellung und kann dort nur auf die Kommentare, welche im Code hinterlegt wurden zurückgreifen.

Fazit [KNIME](#) bietet bereits eine sehr gute Grundarchitektur an, welche im Code selber auch gut Dokumentiert ist. Allerdings bewirkt die fehlende Online Dokumentation, dass es schwierig ist, die Funktionalität um mehr als nur die Standard Anwendungsfälle zu erweitern. Des weiteren bietet [KNIME](#) sehr viele Funktionen dem User an, welche ihn bereits überfordern könnten, wenn er einfach nur ein Diagramm zeichnen möchte. Da sich die Entwicklung vermutlich als mühsam herausstellt und die [KNIME Analytics Platform](#) nicht den Vorstellungen von [simBPMN](#) gerecht wird, ist dies vermutlich keine Applikation welche weiter verfolgt werden sollte.

Core2D

Beschreibung Core2D ist eine .Net Applikation welche Avalonia für das UI einsetzt. Dieses ist ebenfalls nur ein Zeichnungstool und bietet keine BPMN oder andere Funktionalität an.

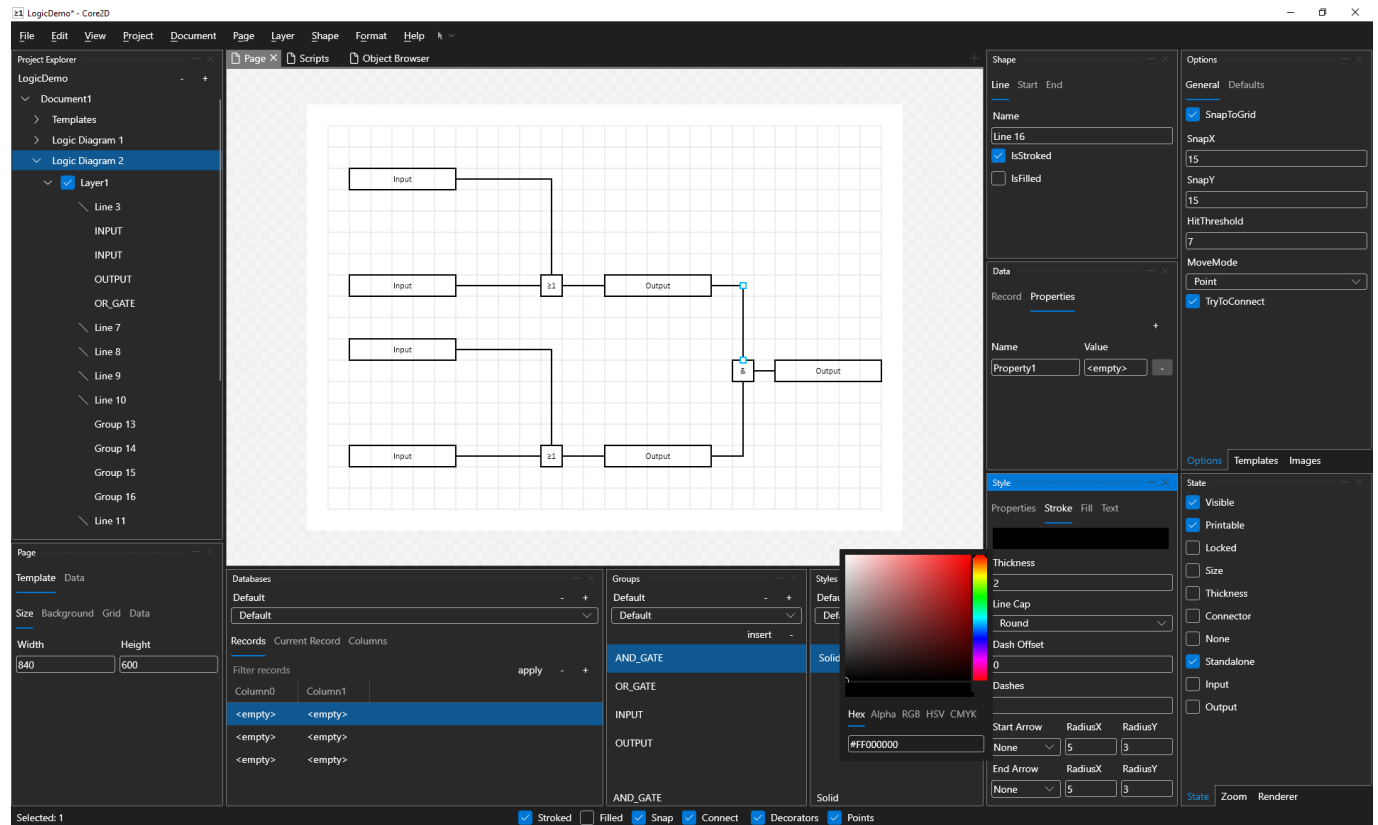


Bild 2.6: Core2D UI

Screenshots

Pro

- Plattformunabhängig
- Benutzt Avalonia UI

Contra

- Keine BPMN Funktionalität geboten

Referenzen

- [Core2D GitHub](#)

Workflowengine

Beschreibung Workflowengine ist ein in .Net Core geschriebene Anwendung welche es ermöglicht Workflows abzubilden. Diese Anwendung ermöglicht es auch eigene Server zu erstellen auf welchen sich die Clients verbinden um Workflows zu zeichnen.

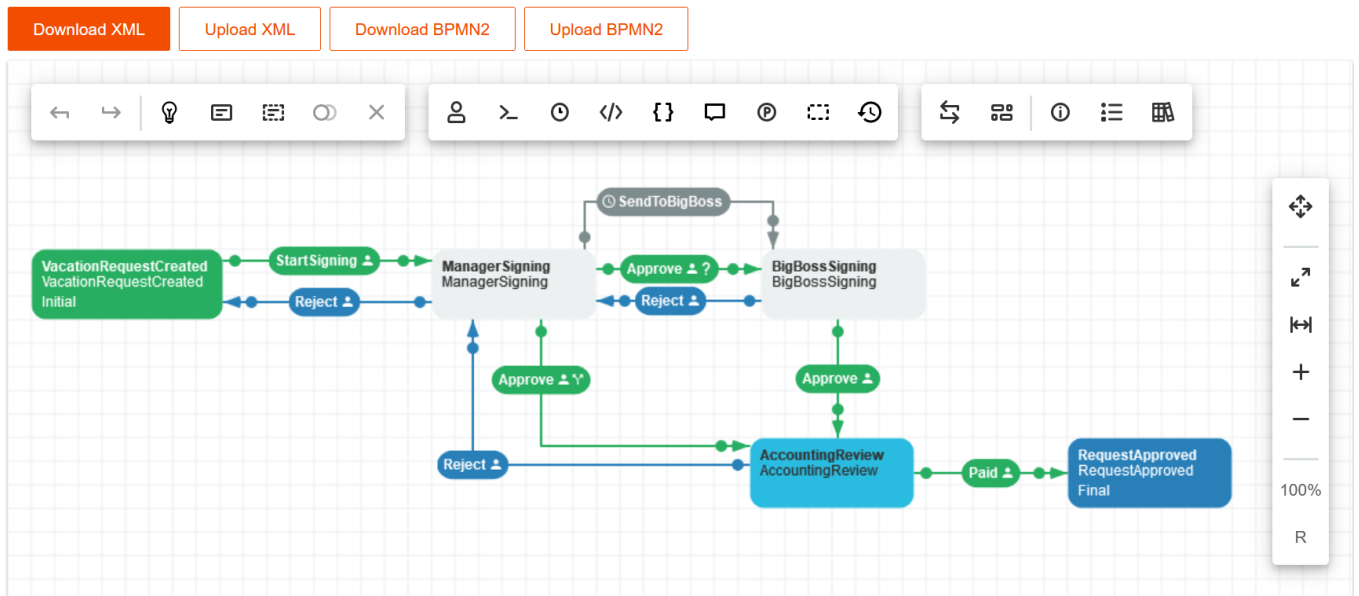


Bild 2.7: Workflowengine UI

Screenshots

Pro

- Plattformunabhängig

Contra

- Kein BPMN Support

Referenzen

- [Workflowengine](#)
- [Workflowengine GitHub](#)

Elsa

Beschreibung Elsa ist ein Workflow manager komplett geschrieben in ASP .Net Core. Es ist darauf ausgelegt Workflows aller Art zu unterstützen. Dabei können ebenfalls auch eigene Activities definiert werden. Erstellte Workflows können in Elsa ausgeführt werden.

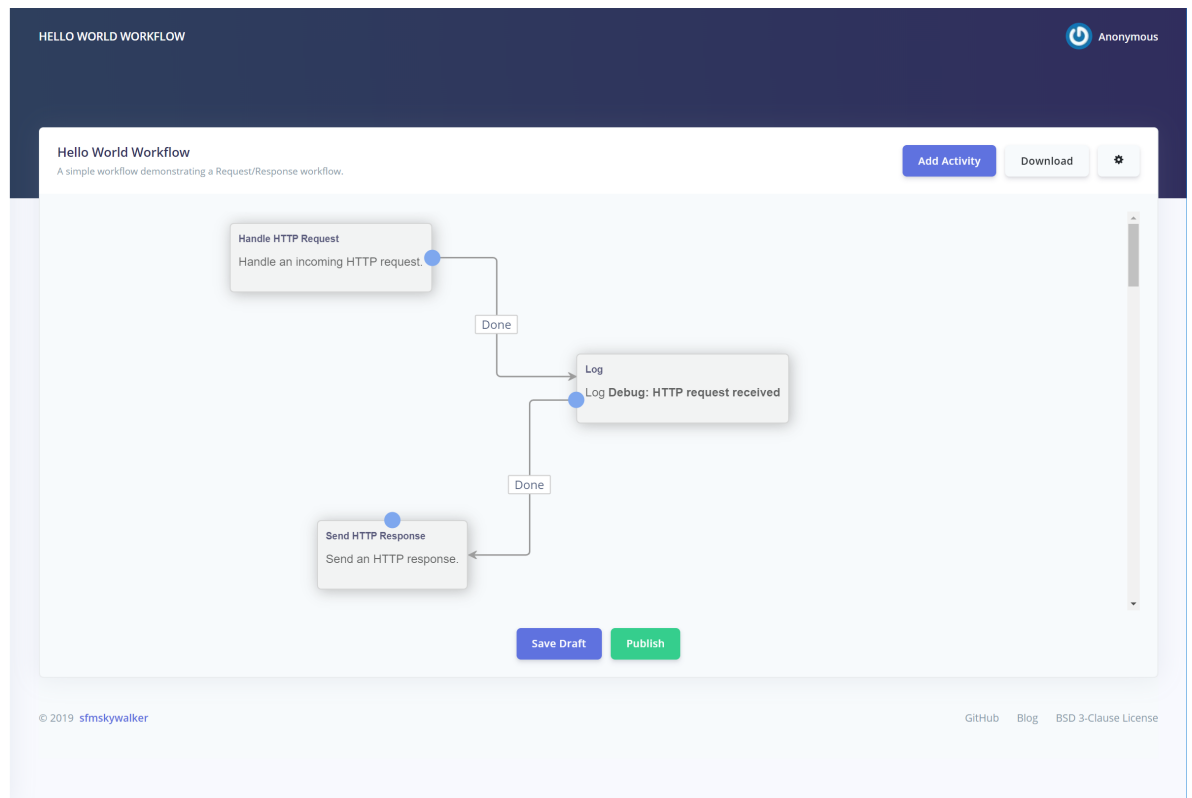


Bild 2.8: Camunda Modeler UI

Screenshots

Pro

- Plattformunabhängig
- Erweiterbar
- Vertraute Technologien
- Workflows sind ausführbar

Contra

- keine BPMN Unterstützung

Referenzen

- [Elsa](#)
- [Elsa GitHub](#)

Slickflow

Beschreibung Slickflow ist ein BPMN Modeler geschrieben in C# .Net Core. Es ist in der Lage BPMN Prozesse abzubilden über einen HTML 5 Editor. Darüber hinaus existieren viele Schnittstellen zu anderen Systemen wie zum Beispiel Datenbanken.

Pro

- Plattformunabhängig
- BPMN-fähig
- Erweiterbar
- Multi-Instanz (Mehrere Benutzer können gleichzeitig am gleichen Task arbeiten)

Contra

- Dokumentation in Chinesisch

Referenzen

- [Slickflow GitHub](#)

Smartflow Sharp

Beschreibung Smartflow Sharp ist ein Workflow Manager geschrieben in C#.

Pro

- Plattformunabhängig

Contra

- Keine BPMN Funktionalität geboten
- Dokumentation in Chinesisch
- Alles ausserhalb von Base Use Cases ist nicht implementierbar (Bsp. custom events)

Referenzen

- [Smartflow Sharp GitHub](#)

Viewflow

Beschreibung Spiffworkflow ist eine weitere Workflow-Engine geschrieben in Python. Diese ermöglicht es BPMN Prozesse abzubilden. Der Anwender kann bequem über eine in Django geschriebene Benutzeroberfläche einen Prozess abbilden.

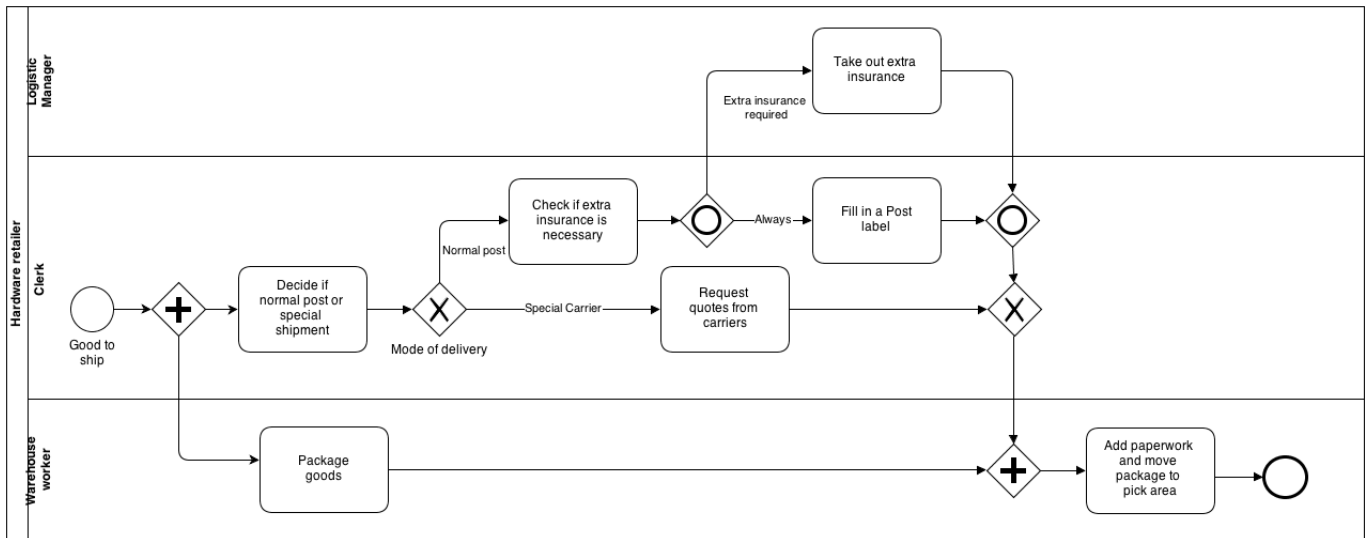


Bild 2.9: Viewflow BPMN Diagram

Screenshots

Pro

- Bietet BPMN Support an
- Plattformunabhängig

Contra

- Code konnte nicht lauffähig gemacht werden

Referenzen

- [Viewflow GitHub](#)

2.5.5 Anwendersoftware

Die folgenden Tools kombinieren alles oben genannte zusammen in eine All-in-One Anwendung. Sie bilden eine optimale Basis um simBPMN Erweiterungen zu implementieren.

BPMN IO

Beschreibung BPMN IO ist ein webbasiertes Tool für BPMN, DMN und Forms. Das in Javascript umgesetzte Tool kann sehr simpel in Anwendungen integriert werden, beispielsweise in Node JS Apps.

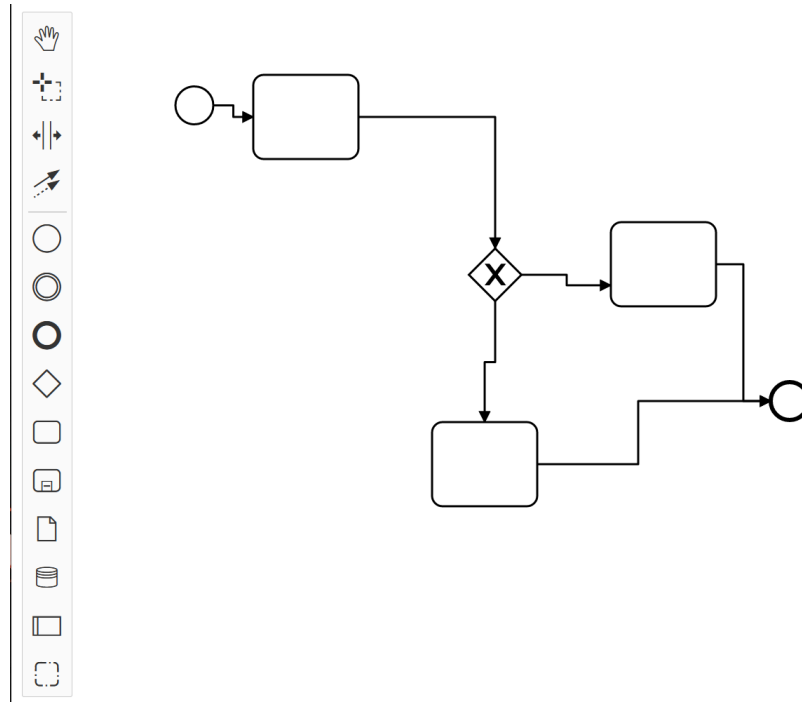


Bild 2.10: BPMN IO UI

Screenshots

Pro

- Plattformunabhängig
- BPMN Funktionalität geboten
- Erweiterbar mit eigenen Items

Contra

- "Nur" ein Modeler, keine weitere Funktionalität

Referenzen

- [BPMN IO](#)
- [BPMN IO GitHub](#)

BPMN Studio

Beschreibung BPMN Studio ist eine Desktop- und Web-Applikation zum Erstellen, Verwalten, Ausführen und Auswerten von BPMN-Prozessen. Das Tool wurde mittels Electron, Aurelia und TypeScript implementiert. Einfach ausgedrückt ist dies eine Node JS Applikation die mittels Electron lokal auf dem Computer ausgeführt wird.

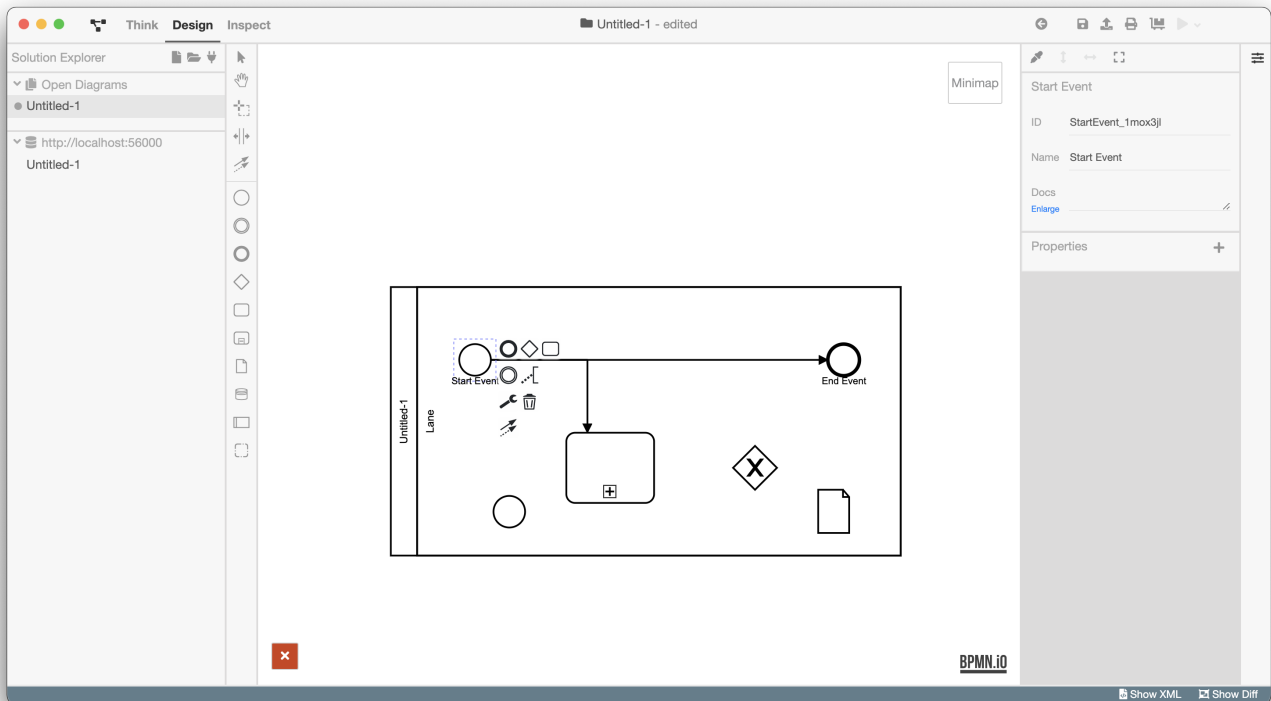


Bild 2.11: BPMN IO UI

Screenshots

Pro

- Plattformunabhängig
- BPMN Funktionalität geboten
- benutzt BPMN IO für die Darstellung der Diagramme
- kann als Docker Image verwendet werden

Contra

- Keine

Referenzen

- [BPMN Studio GitHub](#)

Camunda Modeler

Beschreibung Camunda Modeler ist eine Modellierungs-Software welche BPMN IO verwendet um Diagramme darzustellen. BPMN IO ist ebenfalls aus dem Hause Camunda, daher macht es auch Sinn dass sie ihre eigenen Komponenten in anderen Produkten einsetzen. Der Modeler ergänzt das Diagramm mit Properties die individuell pro Element erstellt werden können. Das ganze ist sehr Modular, selbst das UI kann ausgetauscht werden. Cammand wäre beispielsweise ein anderes UI für den Modeler.

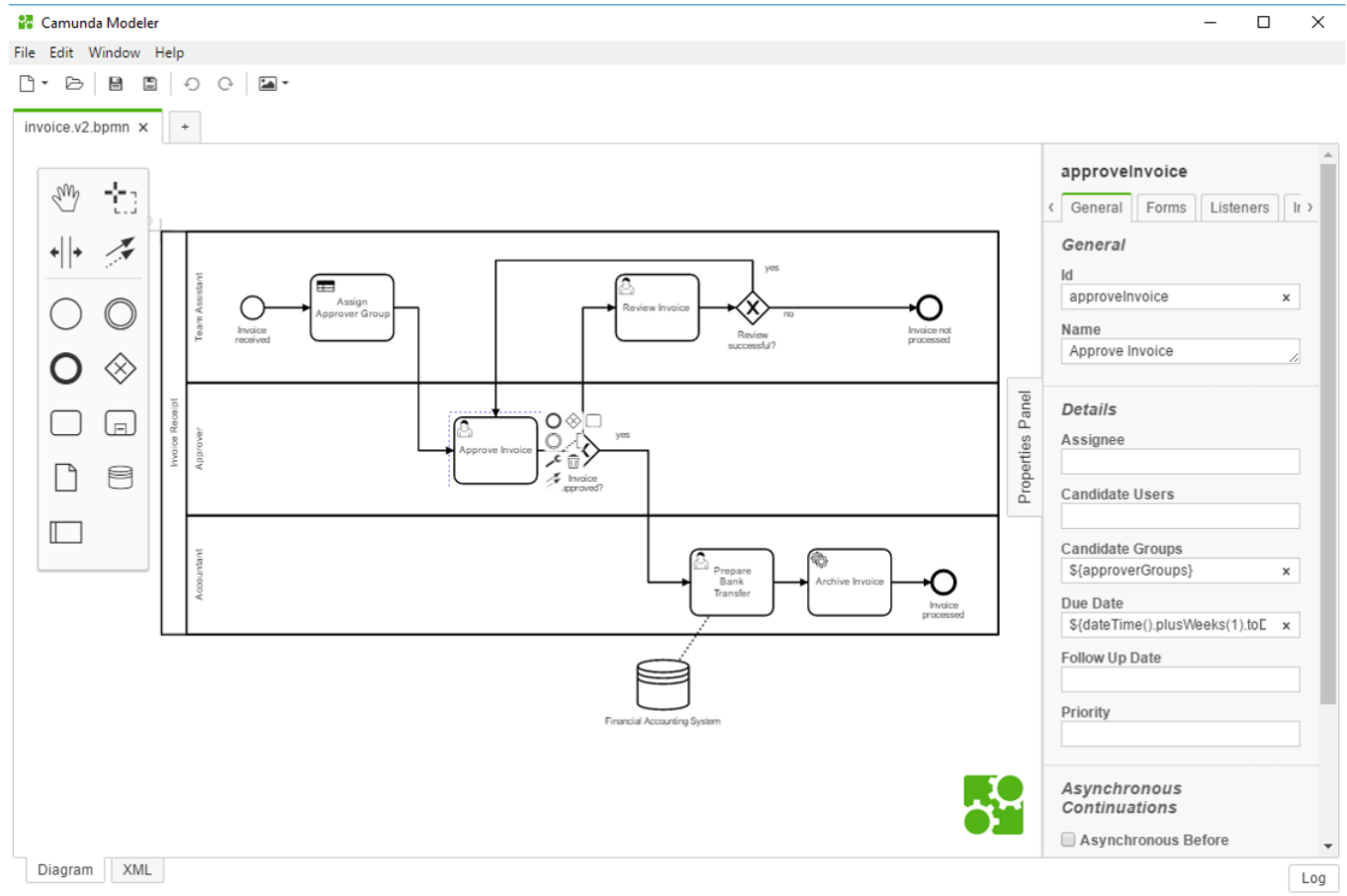


Bild 2.12: Camunda Modeler UI

Screenshots

Pro

- Plattformunabhängig
- BPMN Funktionalität geboten
- Custom Properties
- benutzt BPMN IO für die Darstellung der Diagramme

Contra

- Keine

Referenzen

- [Camunda Modeler](#)
- [Camunda Modeler GitHub](#)
- [Cammand GitHub](#)

2.5.6 Weitere Tools

BPMN View

Beschreibung BPMN View sieht auf den ersten Blick aus wie ein Editor, kann aber in Wahrheit nur Diagramme anzeigen und diese in Bilder konvertieren.

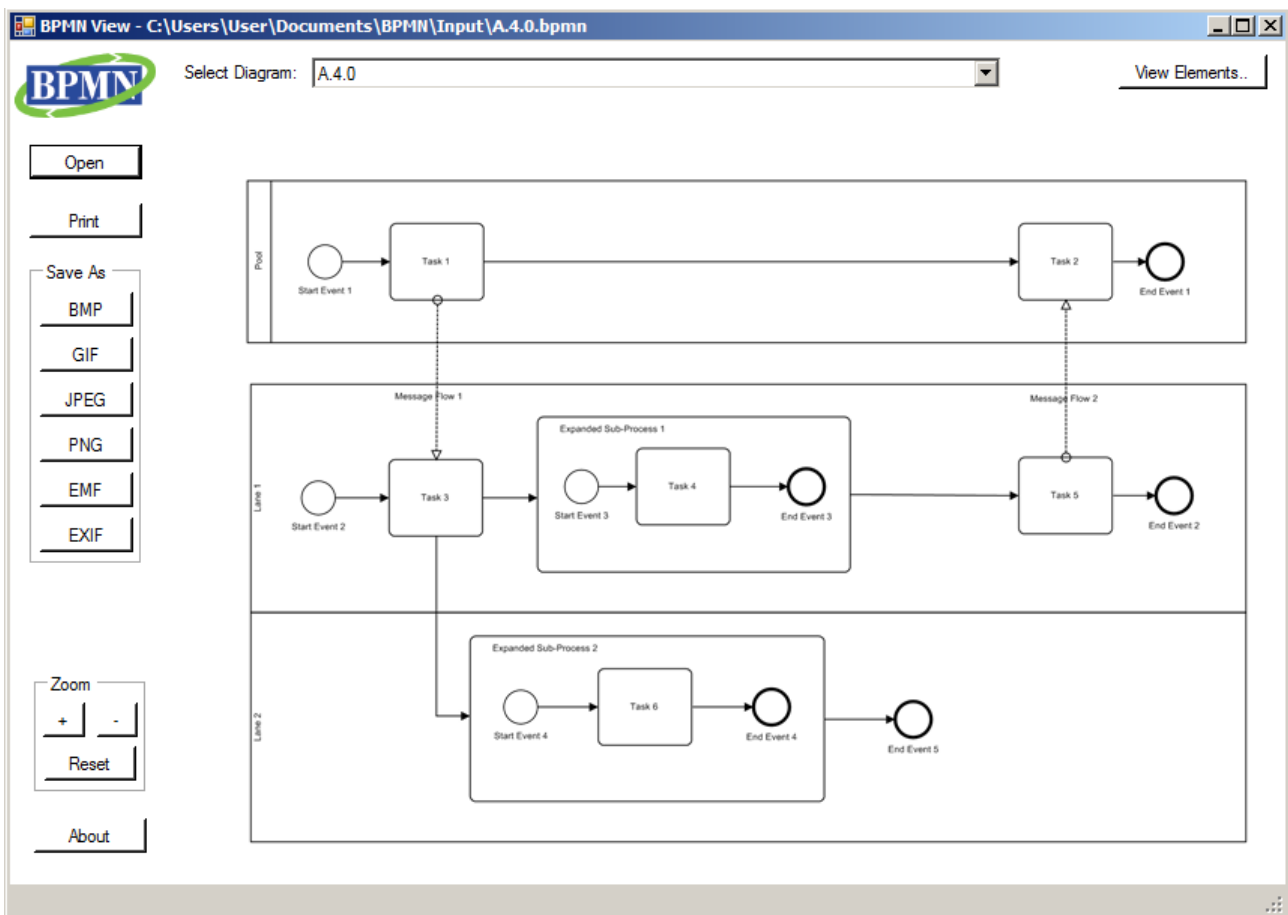


Bild 2.13: BPMN View UI

Screenshots

Pro

- Keine

Contra

- Bietet keinerlei Funktionalität für BPMN an

Referenzen

- [BPMN View GitHub](#)

2.5.7 Ausgearbeitete Variationen

Beschreibung

Folgend werden nun obig genannte Komponenten verschieden zusammengeführt und getestet. Die aufgeführten Variationen sollen zeigen wie Komponenten zusammengehören und ob diese als gesamtes für unsere Zwecke funktionieren. Man beachte dazu dass nicht alle genannten Komponenten aufgelistet sind, da manche durch unsere Killer-Kriterien ausgeschlossen sind, mehr dazu im Fazit.

Variation 1: KNIME

KNIME ist ein Alleskönner im Bereich von Workflows. Es ist auf Cross-Platform Technologien aufgebaut und benötigt deshalb keinen Cross-Platform enabler. Die Anwendung ist schnell erlernbar und bringt keine Mühe für den Anwender. Eine [BPMN 2.0](#)-Unterstützung wird leider nicht von Haus aus geboten und muss selbst implementiert werden. Zur Validierung und Ausführung muss ein Framework wie BPMN RPA eingesetzt werden. KNIME ist zu einem gewissen Grad erweiterbar und kann den Umständen angepasst werden.

Variation 2: Elsa

Elsa ist ein Web-Basiertes Tool um Workflows abzubilden. Da dieses im Webbrowser läuft, sind keine Cross-Platform enabler nötig. Es kann ohne Restriktion erweitert und angepasst werden. Die mitgelieferte Benutzeroberfläche lässt ein wenig zu Wünschen übrig, diese kann allerdings ebenfalls bearbeitet werden. Elsa bietet zur Zeit keine Unterstützung für BPMN, dies muss selbst implementiert werden. Zur Validierung wird darum ebenfalls ein Framework wie BPMN RPA zum Einsatz kommen.

Variation 3: BPMN IO

BPMN IO ist eine Weblösung, entwickelt von Camunda, um BPMN Prozesse darzustellen. Auch hier sind keine Cross-Platform enabler nötig, da das Tool direkt im Browser funktioniert. Unserer Meinung nach bietet BPMN IO die beste Benutzeroberfläche im Aspekt der Bedienbarkeit. Bezüglich Erweiterungen ist es möglich das Tool anzupassen, jedoch müsste dann die Seite selber gehostet werden. BPMN IO ist imstande BPMN zu validieren und auszuführen.

Variation 4: BPMN Studio

BPMN Studio ist aufgebaut auf BPMN IO und bietet darum alle relevanten Funktionen davon an. Im Unterschied zu BPMN IO wird BPMN Studio direkt auf dem Client lokal ausgeführt und benötigt keine Internetverbindung. Als Komplettlösung benötigt es keine weiteren Komponenten mehr um benutzt werden zu können.

Variation 5: Camunda Modeler

Camunda Modeler ist ebenfalls auf BPMN IO aufgebaut und wird lokal auf dem Client ausgeführt. Dies ist ebenfalls eine Komplettlösung und benötigt keine weiteren Komponenten.

Variation 6: Eigenentwicklung

Bei dieser Variante beginnt man auf der grünen Wiese. Alles muss selbst implementiert werden, dafür erhält man den höchsten Grad an Customization und Erweiterbarkeit. Das Resultierende Produkt wäre am Schluss perfekt auf unseren Use Case angepasst. Eine solche Lösung könnte wie folgt aussehen: Das UI macht man mittels einem Cross-Platform enabler wie z.B Avalonia. Die gezeichneten BPMN Prozesse werden durch ein Python Framework wie BPMN RPA validiert oder ausgeführt.

2.5.8 Fazit

Beschreibung

Die unten ersichtliche Tabelle soll zeigen wie der Entscheidungsprozess stattgefunden hat. Diese beinhaltet nur Einträge von den vorherig Evaluieren Konzepten.

Name	BPMN	Lokal	Erweiterbar	Intuitiv	Cross-Platform
BPMN IO	✓	✗	✓	✓	✓
BPMN Studio	✓	✓	✓	✓	✓
Camunda Modeler	✓	✓	✓	✓	✓
Eigenentwicklung	✗	✓	✓	✓	✓
Elsa	✗	✗	✓	✗	✓
KNIME	✗	✓	✓	✓	✓

Tabelle 2.6: Evaluation tabellarische Übersicht

Killer Kriterien

Alle vorgestellten Variationen haben ihre eigenen Vor- und Nachteile. Um aber eindeutig einen "Gewinner" zu definieren benötigen wir Killer Kriterien. Wir haben dazu folgende definiert:

- Das Rad wird nicht neu erfunden
- Applikation läuft lokal auf dem Rechner
- Applikation ist Intuitiv
- BPMN wird unterstützt
- Applikation ist Cross-Platform fähig

Schlussfolgerung

In einer ersten Auswahl haben wir uns für KNIME entschieden, da dieses auf dem Computer direkt ausgeführt werden kann und ähnliche Funktionen zum darstellen von Diagrammen bietet. Während der Implementation haben wir bemerkt dass unsere Use Cases zwar möglich wären umzusetzen, die Handhabung der Anwendung jedoch beträchtlich erschwert wurde. Zusammen mit Prof. Dr. Andreas Rinkel wurde anschliessend besprochen und festgelegt einen andere Lösung zu suchen, bei welcher die Anwender unserer Applikation nicht so sehr leiden müssen. In einem zweiten Schritt haben wir deshalb unsere Suchkriterien an eine Software ein wenig gelockert und auch nach Web-Basierten Tools ausschau gehalten. Um möglichst einfach und schnell eine gute Basis zu erstellen, entschieden wir uns für eine Komplettlösung, dafür kamen die beiden Tools "BPMN Studio" und "Camunda Modeler" in Frage. Glücklicherweise können beide Applikationen lokal und ohne Internet ausgeführt werden. Ein kurzer Vergleich beider Tools zeigte, dass der "Camunda Modeler" eine bessere Unterstützung bietet für unsere Zwecke. Wir haben uns schliesslich dann für dieses Tool entschieden.

Kapitel 3

Projektplan

3.1 Projektorganisation

3.1.1 Organisationsstruktur



	Verantwortung	Entwicklung, Analyse, Architektur
	Name	Mirsayyah
	Vorname	Michel
	OST-Email Adresse	michel.mirsayyah@ost.ch
Erfahrung		Softwareentwicklung .NET und Java
	Verantwortung	Entwicklung, Dokumentation (Zeiterfassungskontrolle, Design, Protokolle)
	Name	Höpfner
	Vorname	Sven
	OST-Email Adresse	sven.hoepfner@ost.ch
	Erfahrung	

Tabelle 3.1: Projektorganisation und Verantwortung

3.1.2 Externe Schnittstellen

Funktion	Name
Betreuer	Prof. Dr. Andreas Rinkel
Experte	Knut Schmahl

Tabelle 3.2: Externe Personenschnittstellen

3.2 Managementabläufe

3.2.1 Zeitbudget

Dem Modul "Studienarbeit" werden 8 ECTS-Punkte angerechnet. Dies entspricht umgerechnet pro Person ein Budget von 16h pro Woche, welches über 14 Wochen verteilt wird. Somit ergibt sich ein Gesamtbudget von 448 Stunden.

Die Studienarbeit wird während des Herbstsemesters 2021 durchgeführt.

Start-Termin: 20.09.2021 **End-Termin:** 23.12.2021

3.2.2 Zeitliche Planung

Das Projekt wird mit dem erlernten Model **Scrum+** durchgeführt. In diesem werden die Phasen des Modells Rational Unified Process mit **Scrum** kombiniert. Entsprechend werden während der Phasen Sprints von jeweils 2 Wochen durchgeführt. Das Projekt startet mit einer Einführung, welche durch Prof. Dr. Andreas Rinkel durchgeführt wird und Ended mit einer Abschlusspräsentation. Innerhalb dessen können für die 14 Wochen somit 7 Meilensteine definiert und geplant werden.

simBPMN Toolanalyse Zeitplan

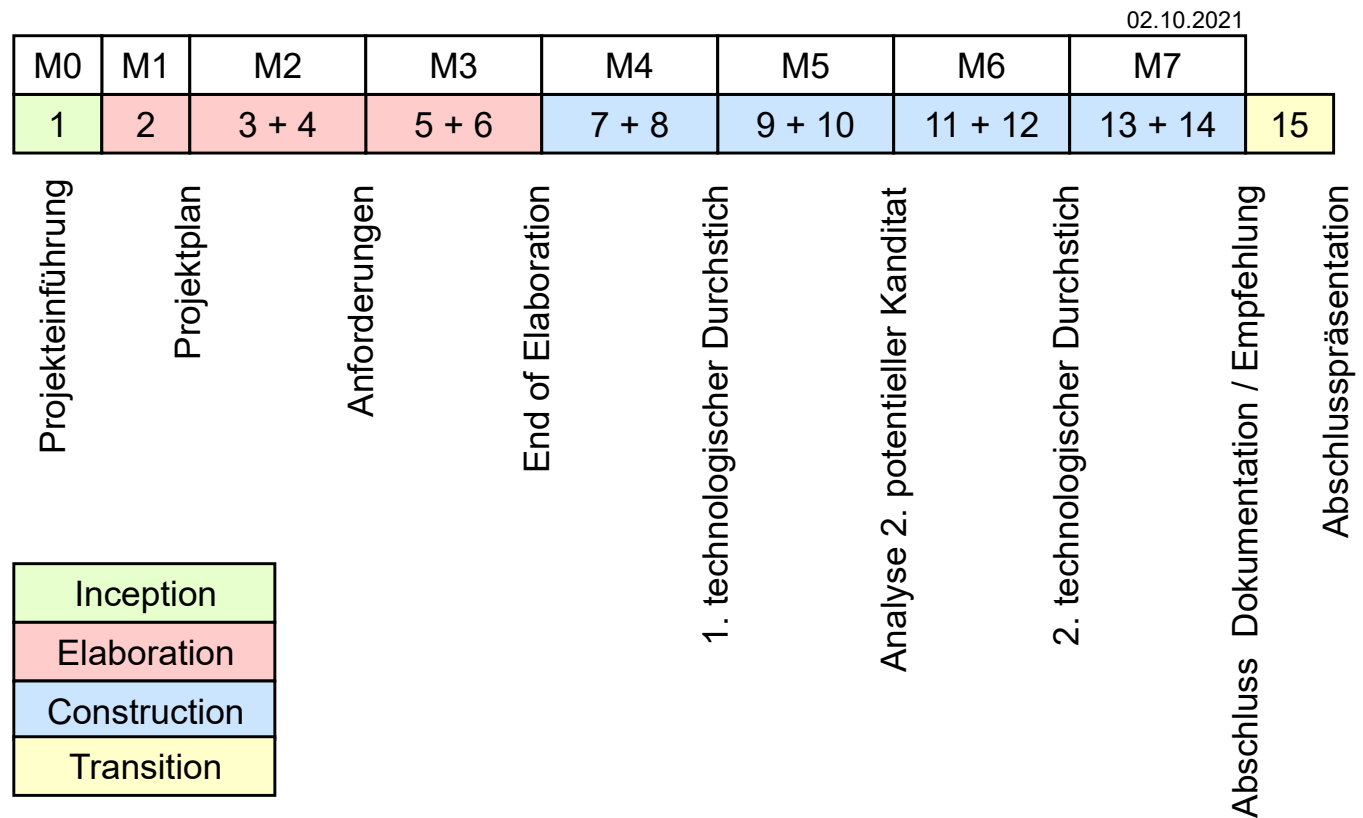


Bild 3.1: simBPMN Zeitplan

Meilensteine

Nummer	Bezeichnung	Erreicht am	Beschreibung
M0	Projekteinführung	24.09.2021	Projektauswahl sowie Projekteinführung mit dem Betreuer zum Thema simBPMN
M1	Projektplan	30.09.2021	Der Projektplan wurde erstellt und mit dem Betreuer abgesprochen.
M2	Anforderungen	14.10.2021	Die Anforderungen für die Arbeit wurden definiert und in der Anforderungsspezifikation festgehalten.
M3	End of Elaboration	28.10.2021	Die vielversprechendsten Kandidaten wurden oberflächlich beurteilt und ein Potentieller Kandidat ausgewählt, welcher weiter analysiert wird.
M4	1. technologischer Durchstich	11.11.2021	Für den ersten Kandidaten wurde ein Durchstich erstellt. Durch diesen kann beurteilt werden, ob sich dieser für weitere Arbeiten eignet.
M5	Analyse 2. potentieller Kandidat	25.11.2021	Der Kandidat welcher nach dem ersten Kandidat am vielversprechendsten aussieht, wird ausgewählt und weiter analysiert werden.
M6	2. technologischer Durchstich	09.12.2021	Für den zweiten Kandidaten wurde ein Durchstich erstellt. Durch diesen kann beurteilt werden, ob sich dieser für weitere Arbeiten eignet.
M7	Abschluss Dokumentation	23.12.2021	Die erarbeiteten Erkenntnisse wurden zu diesem Zeitpunkt in die Dokumentation eingearbeitet und diese wurde für die Abgabe überarbeitet.
M8	Abschlusspräsentation	18.02.2021	Es wurde eine Präsentation vorbereitet und abgehalten, in welcher die Ergebnisse der Arbeit präsentiert wurden.

Tabelle 3.3: Meilensteine

Sprints

Die Dauer eines Sprints ist auf zwei Wochen definiert und stimmt somit mit den Meilensteinen überein. Der Übergang eines Sprints findet somit immer am Donnerstag statt. Die Ziele der verschiedenen Sprints sind somit an die der Meilensteine geknüpft.

Sonstiges

Während der gesamten Projektdauer wird die Dokumentation entsprechend erweitert und aktualisiert. Der Stand dieser wird am Ende jedes Sprints überprüft.

Bei allfälligen Problemen oder Besonderheiten wird dies in der wöchentlichen Besprechung besprochen und im Meetingprotokoll festgehalten.

3.2.3 Besprechungen

Es werden wöchentliche teaminterne Meetings abgehalten, diese entsprechen dem Weekly [Scrum](#). An diesen Meetings werden der aktuelle Stand der Arbeit, mögliche Schwierigkeiten, sowie die bevorstehenden Punkte ausgetauscht und besprochen.

Zusätzlich wird wöchentlich ein Meeting in Person mit dem Team und dem Betreuer auf dem Gelände der OST abgehalten. In diesem wird der Stand sowie das weitere Vorgehen besprochen. Die Besprechungen werden protokolliert. Auf diese Meetings folgen jeweils teaminterne Meetings, welche als Sprint Planning, Retrospective sowie als Refinement Meeting dienen.

Ausserhalb der geplanten Meetings sind beide Gruppenmitglieder, sowie der Betreuer offen für zusätzliche Besprechungen, sollten diese von Nöten sein.

3.3 Risikomanagement

Die Risiken des simBPMN Visualizers halten sich in Grenzen, da keine Abhängigkeiten zu anderen Systemen existieren. Durch die plattformunabhängigen eingesetzten Technologien bestehen zwar keine Abhängigkeiten zu Betriebssystemen, jedoch zur Technologie beziehungsweise dem Framework welches die plattformunabhängigkeit ermöglicht. Die eingesetzte [simBPMN](#) Notation ist bereits definiert und muss nicht neu erstellt werden. Alle weiteren wichtigen Risiken sind unter "[Tabelle 3.3.1. Risikoanalyse](#)" zu finden.

3.3.1 Risikoanalyse

Mit der Risikoanalyse wird versichert, dass der simBPMN Visualizer für bekannte Risiken über Strategien verfügt, um mit diesen umzugehen. Unter anderem wird das Vorbeugen, sowie das Reagieren auf Risiken geschildert. Durch diese Risikoanalyse wird einem Scheitern des Projekts entgegengewirkt.

- Projekt: simBPMN Visualizer
- Erstellt am: 27.09.2021

Nr	Risiko	Beschreibung	max. Schaden	Eintrittswahrscheinlichkeit	Vorbeugung	Verhalten beim Eintreten
R1	Das Untersuchte Tool oder Framework wird deprecated	Das Tool oder Framework wird nicht mehr weiterentwickelt oder unterstützt	60h	1%	Keine	Das Tool oder Framework wird entweder verworfen oder selber weiterentwickelt, sollte sich die Möglichkeit dazu ergeben.
R2	simBPMN beinhaltet konzeptionelle Fehler	Im Konzept werden Fehler entdeckt, die nicht implementiert werden können	10h	3%	Keine	Ausarbeitung der Fehler zusammen mit den Stakeholdern
R3	Krankheit	Teammitglieder fallen wegen Krankheit aus	32h	5%	Arbeiten protokollieren	Dokumentation wird aktuell gehalten. Ausgefallene Teammitglieder werden auf den neusten Stand gebracht.

Tabelle 3.4: Risikomanagement

3.3.2 Umgang mit Risiken

Qualitätssichernde Überprüfungen: Nach jedem Sprint werden die Risiken neu evaluiert, sodass eine hohe Qualität des Risikomanagements während der gesamten Projektdauer garantiert werden kann.

3.4 Arbeitspakete

Arbeitspakete umfassen Epics, User Stories und Tasks. Sie werden als Issues in [YouTrack](#) erfasst und durch Labels unterschieden. Die Issues werden stets verfeinert und aktuell gehalten. Die Issues werden auf beiden Projekten, Code: [SA](#) und Dokumentation: [SA-Dokumentation](#), geführt. Der gesamte Produkt-Backlog setzt sich demnach aus allen Issues der beiden Projekte zusammen und befindet sich im Hauptprojekt: [Gruppe SA](#).

3.5 Infrastruktur

- [GitLab](#): Versionierung, Backlog
- [YouTrack](#): Zeiterfassung
- \LaTeX -Editor: Jedes Mitglied ist in der Wahl seines Editors frei

3.6 Qualitätsmassnahmen

Massnahme	Projekt	Beschreibung	Ziel
Gegenseitige Reviews	SA sowie SA-Dokumentation	Jede Zeile der Dokumentation und der Codebase wird von mindestens einem Gruppenmitglied geprüft. Bei Verbesserungsvorschlägen werden diese diskutiert und solange umgesetzt, bis beide Seiten (Autor und Prüfer) einverstanden sind. Nur abgesegnete Änderungen werden in den Master Branch comitted.	Hohe Qualität der Dokumentation und des Codes
Codestyle- und Branching Guidelines	SA	Umfangreiche, klar definierte und dokumentierte Guidelines zu Coding- und Branching	Vereinheitlichung des Codes und der Commit-Historie für das gesamte Repository
Testing in Build	SA	Umsetzen der CI/CD Practice "Make your Build Self-Testing".	Bug-freies und stabiles Produkt
Code Quality Tools	SA	Einsatz von Static Code Analysis Tools.	Einhalten der Code-Style Guidelines; Hohe Qualität des Codes

Tabelle 3.5: Qualitätsmassnahmen

3.6.1 Code Reviews

Im Zuge jedes Merge Requests auf den Master Branch wird ein Code Review der vorgebrachten Änderungen durch mindestens ein Gruppenmitglied durchgeführt. Erst wenn das Code Review zu einem zufriedenstellendem Ergebnis gelangt, wird dem Merge Request stattgegeben.

3.6.2 Guidelines

Klar definierte Guidelines, die zu Projektbeginn definiert werden, sollen die Vereinheitlichung von Codingstyles und Branchingstrategien ermöglichen.

Dokumentation

Die Dokumentation wird mithilfe von Latex geschrieben. Um einen diese Übersichtlich zu behalten, wird für jedes neue Thema ein neues Latex-File erstellt. Somit kann sichergestellt werden, dass zu jedem Thema nur das geschrieben wird, was auch etwas damit zu tun hat und somit keine Verwechslungen entstehen können.

Code Style Guidelines

Als Grundlage gelten die Guidelines der jeweiligen zugrundeliegenden Sprache.

Branching- und Commit Guidelines

Diese Guidelines gelten nur für das Projekt simBPMN Visualizer.

Es wird ein Rebase Workflow verfolgt. Vor jedem Merge Request wird, falls nötig, ein Rebase des lokalen feat/bug Branches auf den Origin Master durchgeführt. Dies simplifiziert die Commit History des Repositories und verbessert damit die Übersichtlichkeit und Nachvollziehbarkeit.

Commits erfolgen regelmässig und umfassen relativ wenige Änderungen. Es wird regelmässig ins Origin Repository gepusht.

Folgende Branches werden genutzt:

Branch	Name	Beschreibung
Stable	master	Repräsentiert stabile Version des Projekts. Commits erfolgen ausschliesslich durch Development Branch.
Development	dev	Repräsentiert aktuellen Entwicklungsstand des Projekts. Commits erfolgen ausschliesslich durch Merge Requests von Feature, Bug oder Hotfix Branches.
Features/Bugfix	feat-* / bug-*	Repräsentiert Arbeit an Feature beziehungsweise Bugfix.
Hotfix	hotfix-*	Repräsentiert Hotfixes für den Stable Branch.

Tabelle 3.6: Branches

Commit Message Guidelines

Commit Messages werden nachfolgendem Muster erstellt: `<type>(<scope>): <subject>`

Beispiel: `refactor(users-list): refactor to generic approach`

Type

feat: Einführung eines Features

fix: Bug fix

style: Änderungen am Frontend

refactor: Refaktorisierung

test: Hinzufügen von Tests

Scope

Spezifizieren des Orts der Änderung, z.B. Name der Klasse oder Methode.

Subject

Kurze und stichhaltige Beschreibung der Änderung in Englisch. In Präsens, kleingeschrieben und ohne Punkt am Ende.

Anforderungsspezifikationen

Die Anforderungen sind massgeblich durch die Bestimmungen der [OST](#) geprägt. Die Hauptanforderung ist es eine Grundlage für ein Tool zu bestimmen, mit welchem ein Prozess modelliert werden kann. Diese Grundlage soll mit der [simBPMN](#) Notation erweiterbar sein. Prozesse der realen Welt sollen so, auf eine einfache Art digital modelliert werden können.

Das Tool soll soweit aufgesetzt werden können, so dass eine einfache Erweiterung später zu einem späteren Zeitpunkt möglich ist.

4.1 Produktfunktionen

Die Grundlage des Tools soll die Möglichkeit besitzen, dieses mit der [simBPMN](#) Notation zu erweitern. Der Anwendungsbenutzer soll dadurch die Möglichkeit erhalten, einen Prozess graphisch zu modellieren.

Die modellierten Objekte müssen mit Eigenschaften editierbar sein, um so eine detaillierte Beschreibung zu erlauben. Diese kann per Text oder auch per Dateianhang erfolgen.

Der Prozess soll in einem sinnvollen Datenformat, wie [JSON](#) oder [XML](#), abgespeichert werden können. Dies soll die Möglichkeit erbringen, dass das modellierte Model auf Fehler oder Probleme untersucht werden kann, welche dem Anwendungsbenutzer dargestellt werden können.

4.2 Lieferumfang

Der Lieferumfang für diese Arbeit ist eine Analyse von verschiedenen Tools und Frameworks, welche auf die beschriebenen Anforderungen und Funktionen untersucht wurden. Mithilfe dieser Analyse wird auch eine Empfehlung ausgesprochen, welches Tool oder Framework sich für eine Umsetzung der Anforderungen voraussichtlich am besten anbietet.

Zusätzlich zu der Analyse enthält der Lieferumfang einen Projektplan, welcher das Vorgehen und Entscheidungsprozesse, während der Arbeit beschreibt und Dokumentiert, ein Kompendium, welches die Erkenntnisse und Vorgehen von verschiedenen Tools beinhaltet und Plakat im A3 Format, welches nach der Arbeit ausgestellt wird.

Diese Punkte sowie diverse Formale Punkte (Eigenständigkeitserklärung, Glossar, Referenzen, usw..) werden in digitaler Form über das Arbeitsverwaltungstool (AVT) der Ost abgegeben.

4.3 Benutzer Charakteristik

Zielgruppe des Produktes:

- Primär: Entscheidungsträger für einen [simBPMN](#) Visualizer

Die Entscheidungsträger können anhand dieser Analyse entscheiden, ob das Tool umgesetzt werden kann und auf welchen der analysierten Tools oder Frameworks sich bezogen wird.

- Sekundär: Entwickler für einen [simBPMN](#) Visualizer

Die Entwickler des Tools können sich auf die Erkenntnisse des Tools und der Plattform sowie auf die definierten Anforderungen für das Tool berufen.

4.4 Annahmen, Einschränkungen und Abhängigkeiten

4.4.1 Annahmen

Es bestehen keine speziellen Annahmen für die Anforderungen.

4.4.2 Einschränkungen

Die Analyse findet nur oberflächlich statt. Es wird überprüft in wie fern die Funktionalitäten umgesetzt werden können. Aufgrund von zeitlichen Einschränkungen werden nur die vielversprechendsten Optionen ausgewählt und im Detail analysiert.

4.4.3 Abhängigkeiten

Die Analyse ist von keiner weiteren Arbeit abhängig.

4.5 Use Cases

4.5.1 Use Case Diagramm



Bild 4.1: Use Case Diagramm

4.5.2 Beschreibungen (Fully Dressed)

Die nachfolgenden Use Cases im Fully Dressed Format entsprechen dem Funktionsumfang, welche ein Tool oder Framework mitbringen sollte, um weiter evaluiert werden zu können.

Prozess Modellierung

Use Case Abschnitt	Beschreibung
ID	1.1
Umfang	Minimal Viable Product
Level	Benutzerinteraktion
Primärer Akteur	Benutzer
Stakeholder und Interessierte	Benutzer: Will einen Prozess im Tool modellieren.
Vorbedingungen	—
Erfolgsgarantie	<ul style="list-style-type: none"> • Der kann Benutzer seinen Prozess erfolgreich der vom Tool gegebenen Notation abbilden.
Hauptszenario (erfolgreich)	<ol style="list-style-type: none"> 1. Benutzer öffnet einen bestehenden oder neuen Prozess. 2. Die Software zeigt dem Benutzer die möglichen Elemente an. 3. Der Benutzer modelliert seinen Prozess mit Drag & Drop Aktionen.
Erweiterungen	—
Spezielle Bedingungen	—
Technologie- und Datenvariationen	—
Häufigkeit des Auftretens	Da dies ein notwendiger Schritt zum Erreichen einer der Kernfunktionalitäten der Software ist, tritt dieser relativ häufig auf.
Verschiedenes	—

Tabelle 4.1: Use Case: Prozess Modellieren

Subprozesse

Use Case Abschnitt	Beschreibung
ID	1.2
Umfang	Minimal Viable Product
Level	Benutzerinteraktion
Primärer Akteur	Benutzer
Stakeholder und Interessierte	Benutzer: Will einen Subprozess abbilden
Vorbedingungen	<ul style="list-style-type: none"> • Der Benutzer hat bereits einen Prozess modelliert.
Erfolgsgarantie	<ul style="list-style-type: none"> • Nach erfolgreichem Öffnen eines bestehenden Prozesses, kann darin ein weiterer Prozess definiert werden.
Hauptzenario (erfolgreich)	<ol style="list-style-type: none"> 1. Der Benutzer öffnet einen bestehenden Prozesses. 2. Der Benutzer wählt das Subprozess Element in der Toolbox aus und zieht dieses in den bestehenden Prozess. 3. Die Software bettet das Element in der Zeichnung ein und ermöglicht die Modellierung des Subprozesses.
Erweiterungen	—
Spezielle Bedingungen	<ul style="list-style-type: none"> • Dieser UseCase ist nur in einem bestehenden Prozess möglich. Ein Subprozess kann nicht für sich alleine stehen.
Technologie- und Datenvariationen	—
Häufigkeit des Auftretens	Da dies ein notwendiger Schritt zum Erreichen einer der Kernfunktionalitäten der Software ist, tritt diese relativ häufig auf.
Verschiedenes	—

Tabelle 4.2: Use Case: Nested Prozesse

Notizen und Anhänge

Use Case Abschnitt	Beschreibung
ID	1.3
Umfang	Minimal Viable Product
Level	Benutzerinteraktion
Primärer Akteur	Benutzer
Stakeholder und Interessierte	Der Benutzer: <ul style="list-style-type: none"> • will einen Notiz hinterlegen. • ein File hinterlegen.
Vorbedingungen	<ul style="list-style-type: none"> • Der Benutzer hat bereits einen Prozess welcher dieser beschreiben möchte.
Erfolgsgarantie	<ul style="list-style-type: none"> • Notizen und Files können direkt aus der Zeichnung heraus aufgerufen werden.
Hauptszenario (erfolgreich)	<ol style="list-style-type: none"> 1. Der Benutzer wählt ein Kommentar- oder Fileobjekt aus der Toolbox aus. 2. Das entsprechende Objekt kann zum beschreibenden Objekt angehängt werden. 3. Files können von der Zeichnung aus geöffnet werden
Erweiterungen	—
Spezielle Bedingungen	—
Technologie- und Datenvariationen	—
Häufigkeit des Auftretens	Da dies eine der Kernfunktionalitäten der Software ist, tritt diese häufig auf.
Verschiedenes	—

Tabelle 4.3: Use Case: Remarks und Files hinterlegen

Element Palette bearbeiten

Use Case Abschnitt	Beschreibung
ID	1.3
Umfang	Minimal Viable Product
Level	Entwicklerinteraktion
Primärer Akteur	Entwickler
Stakeholder und Interessierte	Der Entwickler: <ul style="list-style-type: none"> • Will die Element Palette bearbeiten um Items hinzuzufügen oder zu entfernen.
Vorbedingungen	<ul style="list-style-type: none"> • Keine
Erfolgsgarantie	<ul style="list-style-type: none"> • Es können die Elemente der simBPMN in einem Prozess dargestellt werden.
Hauptszenario (erfolgreich)	<ol style="list-style-type: none"> 1. Der Entwickler hat eine Übersicht über die verschiedenen Elemente. 2. Die Elemente enthalten eigen hinzugefügte Elemente der simBPMN Notation
Erweiterungen	—
Spezielle Bedingungen	—
Technologie- und Datenvariationen	—
Häufigkeit des Auftretens	Da diese Funktionalität nur zur Zeit Entwicklung auftritt, tritt diese einmal zur Entwicklung auf.
Verschiedenes	—

Tabelle 4.4: Use Case: Remarks und Files hinterlegen

4.5.3 Beschreibungen (Brief)

Die nachfolgenden Use Cases im Brief Format sind nicht zwingend erforderlich für die Basisfunktionalität des bestehenden Tools oder Frameworks und besitzen eine entsprechende tiefere Priorisierung. Diese Use Cases sind eher unspezifisch und werden fortlaufend aktualisiert oder gelöscht sowie um neue Use Cases ergänzt.

Validierung eines Prozesses

Dem Benutzer soll es möglich sein, frühestmöglich Fehler in der Modellierung zu erkennen. Unter anderem sind folgende Gründe als Fehler anzusehen:

- Fehlende Start- Endpunkte
- Assoziationen von nicht kompatiblen Objekten
- Loops in Nested Processes

Step by Step Ausführung eines Prozesses

Der Benutzer kann einen modellierten Prozess Step by Step ausführen, damit ist gemeint dass im GUI ersichtlich ist, wo der Prozess steht und was in diesem Zeitpunkt passiert ist. Dies dient dazu die semantische Korrektheit zu garantieren und soll auf keine Art einer Simulation ähneln.

Anbindung (Export) für ein Simulationssystem

Um den abgebildeten Prozess in einem Simulationsumfeld prüfen zu können, soll es möglich sein einen Export des Modells in ein Simulationssystem zu importieren. Alternativ kann auch eine direkte Schnittstelle gesucht werden.

4.6 Nicht-funktionale Anforderungen

Verbindlichkeit

Die Verbindlichkeit der Anforderungen ist wie folgt zu interpretieren:

...muss..., ...ist zu..., ...beträgt...	zwingend, obligatorisch, hart gefordert
...sollte...	empfohlen
...kann..., ...darf...	fakultativ, erlaubt
...darf nicht...	verboten, nicht erlaubt
...muss nicht...	fakultativ, nicht zwingend

4.6.1 Funktionalität

Interoperabilität

Das Tool oder Framework welches zur Entwicklung der Visualisierungssoftware eingesetzt werden soll, muss auf Windows 10 und macOS ausgeführt werden können. Es muss somit beide Systeme unterstützen oder portierbar sein.

4.6.2 Zuverlässigkeit

Wiederherstellbarkeit

Die erstellten Diagramme müssen kontinuierlich vom System gespeichert werden. Das Tool oder Framework sollte somit auf einem Datenfile arbeiten. Wenn dies nicht der Fall ist, muss die Möglichkeit bestehen, das Tool oder Framework um eine solche Schnittstelle zu erweitern.

4.6.3 Effizienz

Ressourcenverbrauch

Die Anwendung muss mit einem 1.5GHz Prozessor und 4 GB Arbeitsspeicher einwandfrei funktionieren.

Performance

Ein Workflow welcher bis zu 100 Elemente enthält sollte innerhalb von 3 Sekunden geladen und angezeigt werden können.

Ein Element kann einem Workflow ohne merkbare Verzögerung hinzugefügt werden.

4.6.4 Benutzbarkeit

Verständlichkeit

Das Tool oder Framework sollte die Möglichkeit bieten, die Sprache auf Deutsch oder Englisch anzuzeigen.

Bedienbarkeit

Die Benutzeroberfläche sollte auf einem 13 Zoll Monitor ohne Probleme platz haben. Die Visualisierung vom Diagramm sollte man entsprechend vergrössern oder verkleinern können.

4.6.5 Sicherheit

Für die Anwendung bestehen keine expliziten Sicherheitsanforderungen.

4.6.6 Wartbarkeit

Analysierbarkeit

Das Tool oder Framework muss eindeutig versioniert sein. Die Versionierung muss nachvollziehbar sein und die einzelnen Versionen sollten einen Changelog beinhalten. Der Quelltext jeder Version muss in einem öffentlichem Versionierungssystem abgelegt sein.

Prüfbarkeit

Das Tool oder Framework sollte bereits Tests enthalten, um somit sicherzustellen, dass die vorhandene Funktionalität keine Bugs aufweist.

4.6.7 Übertragbarkeit

Installierbarkeit

Die Anwendung darf nicht installiert werden. Sie muss direkt auf dem Rechner ausgeführt werden können.

Die Anwendung muss ohne Administratorrechte ausgeführt werden können.

Die Anwendung darf keine Einträge in die Windows Registry schreiben.

Koexistenz

Auf einem Rechner darf nur eine Instanz der Anwendung ausgeführt werden.

Schlussbericht

5.1 Zielerreichung

Das Ziel dieser Studienarbeit was es, kurz zusammengefasst, ein Tool zu finden, welches es ermöglicht Simulationsprozesse mittels [BPMN 2.0](#) zu definieren. Die extensive Evaluation, welche durchgeführt wurde, zeigt Open-Source Projekte, die als Basis für [simBPMN](#) eingesetzt werden können. Ein Grossteil dieser Arbeit war es den Entscheidungsprozess unserer Seite aufzuzeigen und wie wir unsere Kriterien definierten bzw. gewichteten. Zum Zeitpunkt der Studienarbeit ist unserer Meinung nach eine Applikation basierend auf BPMN IO optimal. In einer Folgearbeit kann mithilfe unserer Basis eine Applikation umgesetzt werden, um der Vision von simBPMN ein Stück näherzukommen. Aus unserer Sicht ist der Auftrag der Studienarbeit erfüllt.

Persönliche Erkenntnisse, die während der Arbeit gewonnen wurden, sind im darauf folgenden Kapitel zu finden.

5.2 Review der eingesetzten Tools

GitLab ist ein äusserst umfangreiches und gut geeignetes Tool zur Versionsverwaltung. Da darin auch Issues erstellt werden können, kann GitLab zusätzlich als Issue-Tracker verwendet werden. Der Umgang mit Git wurde bereits in vorgegangen Modulen erlernt und konnte hier ohne weiteres angewendet werden.

Aus dem vergangenen Engineering-Projekt haben wir viel gelernt bezüglich Tools, welche wir für Projekte benötigen. So haben wir beispielsweise unsere Zeiterfassung nicht wie im Engineering-Projekt direkt in GitLab gemacht, sondern in einem separaten Tool namens [YouTrack](#). YouTrack gehört zur JetBrains Familie und kann via Web benutzt werden. Wie auch für andere JetBrains Produkte üblich, ist dieses einfach in der Handhabung und Bedienung. Im Vergleich zum Engineering-Projekt also ein grosses Plus.

Als Issue-Tracker wurde ebenfalls YouTrack verwendet da man so direkt Zeit auf einen Issue buchen kann.

5.3 Persönliche Erfahrungen

In den nachfolgenden Abschnitten erläutert jedes Mitglied seine Erfahrungen während dem Engineering Projekts eigenständig und ohne Korrelation zueinander. Diese Abschnitte sind in der Ich-Perspektive geschrieben, um den persönlichen Aspekt zu verstärken.

5.3.1 Michel Mirsayyah

Vor dem Projektstart

Der Einstieg in die Studienarbeit hat mir gar nicht gefallen. Bei der Auswahl der Themen haben Sven und ich sorgfältig alle verfügbaren Arbeiten angesehen und evaluiert. Es hiess von der Schulleitung, dass man sich auf möglichst viele Arbeiten bewerben soll, um die Zuweisung zu vereinfachen. Hierbei haben Sven und ich uns auf sage und schreibe sieben Themen beworben. Darauf folgend erhielten wir ein Mail der Schulleitung dass alle unsere Themen bereits vergeben wurden und wir uns vom Restpool der verfügbaren Aufgaben erneut entscheiden müssen. Also trafen Sven und ich uns nochmals und machten erneut drei weitere Eingaben. Die Schulleitung erwiderte unsere Präferenzen mit einer Zuweisung einer Arbeit, welche wir uns gar nicht ausgesucht hatten. Von den gesamt 10 eingereichten Eingaben erhielten wir nach Überarbeitungszeit schliesslich eine 11. Arbeit zugewiesen, welche nicht in unserem Interessens- und Fähigkeitsgebiet lag. Ich fand dies nicht in Ordnung und suchte das Gespräch mit einem Verantwortlichen der Schule. Nach mehreren Kontaktversuchen bei welchen man mich gekonnt ignoriert und später abgeschoben hat, erhielt ich endlich einen Termin wo ich mit jemandem reden konnte. Erst nach mehreren Gesprächen mit dem Studienberater und dem Studiengangsleiter wurde klar, dass hier offensichtlich etwas schief gelaufen ist. Ich möchte dazu anmerken, dass ich jeweils meine Freizeit opfern musste um diese Probleme anzugehen, da sich sonst niemand von der Schule zuständig gefühlt hat. Der Studiengangsleiter zeigte aber zum Schluss einen Einsatz und plante einen Termin zur Findung einer geeigneten Studienarbeit.

Allgemeines zum Projekt

Wie vorhin schon erwähnt war der Einstieg relativ mühsam. Das Projekt konnte leider nicht wie bei anderen Gruppen in der ersten Woche am Montag Morgen beginnen, wir mussten in dieser Zeit ein Thema suchen. Prof. Dr. Andreas Rinkel zeigte sich hier sehr kooperativ und versuchte unsere Fähigkeiten einzusetzen. Wir konnten aus einem grossen Repertoire eine passende Arbeit finden mit welcher wir alle zufrieden waren.

Das Projekt

Das Projekt war etwas ganz neues für mich. Ich hatte hier meinen ersten Kontakt mit BPMN. Speziell an dieser Arbeit war für mich der Teil mit der Suche nach möglichen Lösungen. In meinem Berufsleben habe ich so etwas noch nicht direkt angetroffen. Als Entwickler weiss ich meist was und wie etwas zu erledigen ist aufgrund einer Spezifikation. Hier war ich einmal auf der anderen Seite des Tisches und schrieb die Spezifikation im Auftrag der Studienarbeit.

In der ersten Hälfte des Projektes dachte ich dass unser Progress nicht schnell genug ist um etwas vernünftiges auf die Beine zu stellen. Rückblickend stelle ich aber fest dass es gut war eine tiefgehende Evaluation alles verfügbaren Tools zu machen, somit konnte ein Grundstein gesetzt werden auf welchen weiter aufgebaut werden kann.

Persönliches

Auf Zwischenmenschlicher Ebene gab es keine Probleme im Projekt. Wir konnten alle sowohl im Team als auch Autonom arbeiten und hielten uns gut an die Abgemachten Ziele. Ich hatte Spass an der Arbeit trotz Einstiegsschwierigkeiten.

5.3.2 Sven Höpfner

Durch die anfänglichen Schwierigkeiten bei der Studienarbeit hatte ich Probleme mich in diese einzufinden. Ich habe im Verlauf der Arbeit bemerkt, dass ich am Anfang die verschiedenen Probleme falsch verstanden habe und somit auch nicht wusste worauf ich hinarbeite. Im Verlaufe der Arbeit habe ich versucht diese Unklarheiten zu beheben und konnte dies entsprechend aufarbeiten.

Für mich persönlich war das Thema der Studienarbeit (eine Analyse von verschiedenen Frameworks / Tools) eher etwas auf welchem ich mich nicht so wohl fühle. Ich finde es zwar spannend die Technologien zu untersuchen und auszuprobieren, allerdings kann dies auch schnell zu Frust führen. Vor allem finde ich es schwierig dort Dokumentationen für zu erstellen, da man bei solchen Grundlagenarbeiten nicht auf vorherige Arbeiten aufbauen kann.

Auf persönlicher Ebene habe ich keine Probleme mit meinem Partner oder Betreuer feststellen können. Ich hatte das Gefühl von beiden Parteien unterstützt zu werden und konnte mich immer an diese bei Fragen und Problemen wenden.

Generell freue ich mich auf eine Weiterführung der Arbeit und bin gespannt wie weit wir damit kommen.

Kapitel 6

Zeitauswertung

Unten aufgeführt sind Diagramme, welche den verbrauchten Zeitbedarf darstellen. Diese sollen einen aufgeschlüsselten Einblick in die investierte, sowie die geplante Zeit des Projekts geben. Dazu gibt es noch anzumerken dass unser Team nicht wie geplant in Woche 1 mit dem Projekt beginnen konnte. Aufgrund eines Fehlers bei der Zuweisung der Studienarbeitsthemen wurde unser Team zu ein Thema eingeteilt, welches nicht durchgeführt werden konnte. Aus diesem Grund mussten wir zuerst zusammen mit Prof. Dr. Andreas Rinkel in der ersten Woche ein geeignetes Thema finden. Dadurch lässt sich ein kleinerer Zeitaufwand dieser Woche erklären.

6.1 Zeitauswertung pro Person

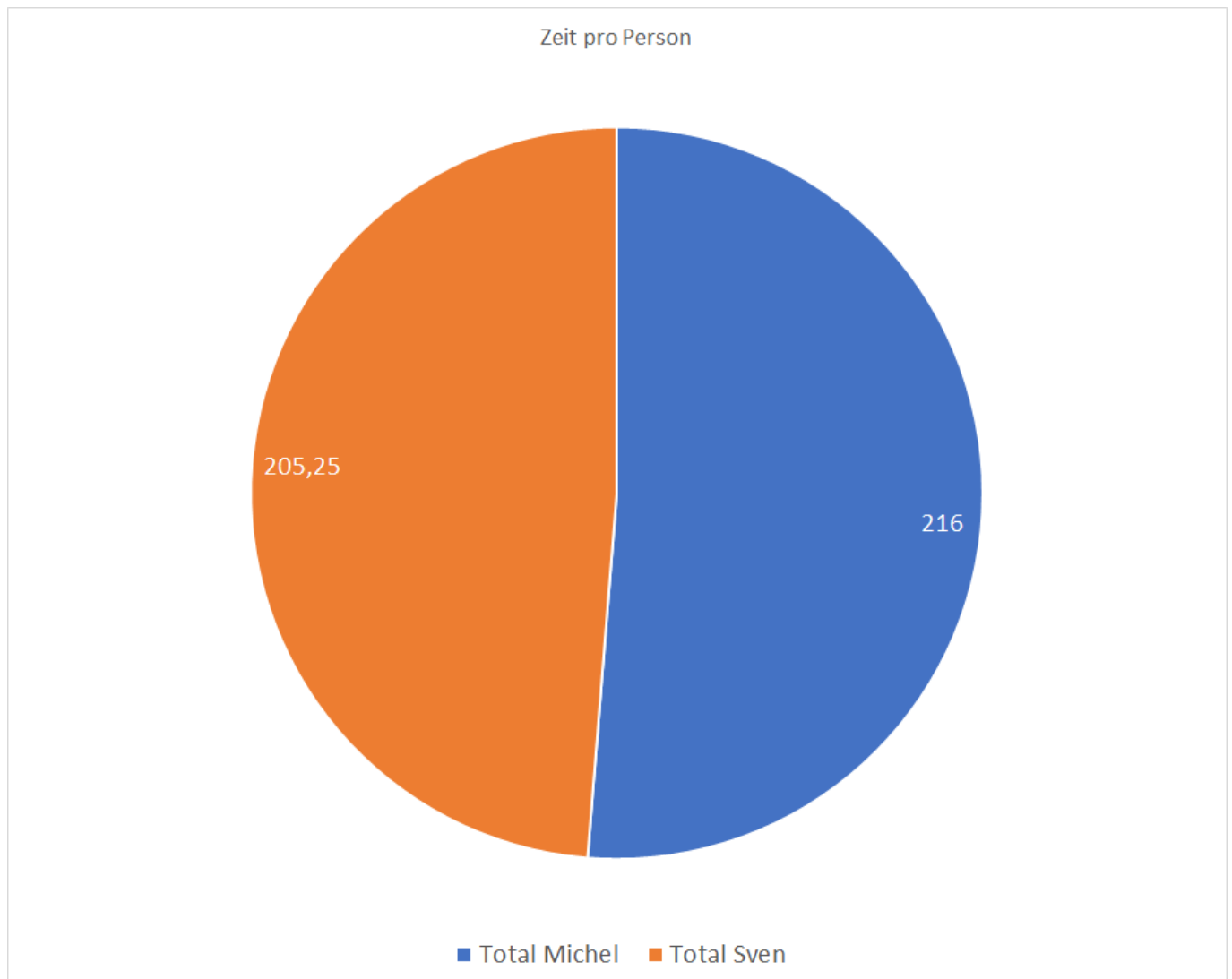


Bild 6.1: Zeitauswertung pro Person

6.2 Zeitauswertung pro Person und Woche

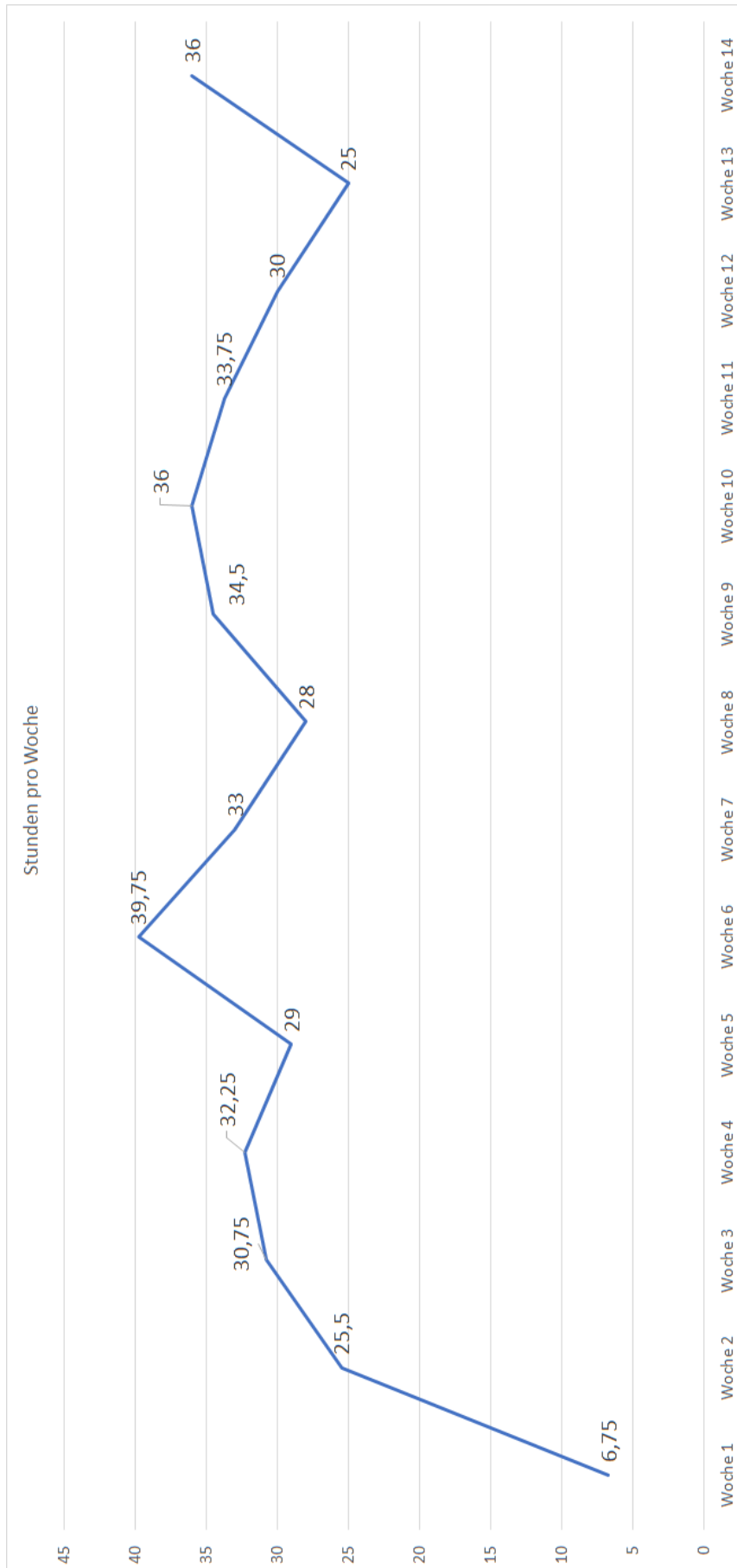


Bild 6.2: Zeitauswertung pro Person und Woche

Kapitel 7

Kompendium

7.1 KNIME

KNIME baut auf **Eclipse** auf und basiert somit auf **Java**. Um eine Erweiterung in KNIME erstellen zu können, müssen folgende Programme installiert werden:

- Java 11 oder höher
- Eclipse IDE for RCP and RAP Developers¹
- Git

Sobald diese Programme eingerichtet wurden, ist es bereits möglich an einer bestehenden Erweiterung zu arbeiten. Wenn man ein neues Projekt aufsetzen möchte, sollte die Dokumentation des KNIME Entwicklerteams befolgt werden.²

7.1.1 KNIME Nodes erstellen

KNIME Nodes werden über den KNIME Node Wizard erstellt. Dieser ist in Eclipse unter *Help* → *Install New Software* verfügbar. Um den Wizard finden zu können muss als Downloadseite <https://update.knime.com/analytics-platform/4.3/> angegeben werden.

Sobald der Wizard installiert wurde, kann dieser über *File* → *New* → *Other...* unter *Create a new KNIME Node-Extension* ausgewählt werden. Im Wizard müssen die folgenden Punkte ausgefüllt sein:

- *New Project Name* **oder** *Existing Project* (Beschreibt das Projekt in welchem der Node erstellt werden soll)
- *Node Class Name* (Beschreibt den Namen des Nodes)
- *Package Name* (Beschreibt das Package in welchem der Node erstellt werden soll)
- *Node Vendor* (Beschreibt den Entwickler / Author)
- *Node Type* (Beschreibt von welchem Typ der Node sein soll)

Wenn man nun den Wizard abschliesst werden folgende Dateien erstellt:

- `MyNodeFactory.java`

In der `NodeFactory` alle Teile eines Nodes zusammengeführt. Die `Factory` steuert die Erstellung des `NodeModels`, des `NodeDialogs` und der `NodeView`. Die `Factory` ist auch die Klasse vom Node, welche bei der Extension angegeben wird.

- `MyNodeModel.java`

Das `NodeModel` beinhaltet die Implementation des Nodes und bestimmt somit was dies für eine Funktionalität bereitstellt. Die Input- und Outputports werden im `NodeModel` ebenfalls definiert.

- `MyNodeDialog.java`

Der `NodeDialog` ist der Dialog, welcher geöffnet wird, wenn man den Node Konfigurieren möchte. Der Dialog wird mithilfe von Java Swing entwickelt. Dies ermöglicht einem den Zugang zu allen Standard Elementen für die GUI Entwicklung.

- `MyNodeView.java`

¹<https://www.eclipse.org/downloads/packages/release/2021-03/r/eclipse-ide-rcp-and-rap-developers>

²<https://github.com/knime/knime-sdk-setup>

Die NodeView ist eine optionale Klasse, welche das Resultat des Nodes anzeigt.

- MyNodeFactory.xml

Das XML-File der NodeFactory beinhaltet die Beschreibung des Nodes sowie verschiedene Metadaten. In diesem File werden auch die Node Icon und der Node Typ angegeben.

- default.png

Das Icon des Nodes wird im Workflow Editor angezeigt. Das Icon des Nodes sollte mit einer passendem Icon, welches die Auflösung 16x16 Pixel beträgt ausgetauscht werden.

- plugin.xml und META-INF/MANIFEST.MF

In diesen Files wird die Extension konfiguriert. Es empfiehlt sich den Standardviewer von Eclipse dafür zu verwenden, da dort bereits die Werte vorgegeben werden. Man kann zum Beispiel dort auch definieren, unter welchem Pfad die Nodes im Workflow Editor verfügbar sind.

7.1.2 KNIME Workflow Editor

Es ist möglich per Hand Workflows im Workflow Editor von KNIME anzupassen. Mit einem Rechtsklick in den Editor wird ein Menu geöffnet, mithilfe welches man folgende Sachen dem Workflow hinzufügen kann:

- Workflow Annotation

Die Workflow Annotation kann benutzt werden, um dem Workflow eine Beschreibung hinzuzufügen.

- Metanode

Ein Metanode ist als ein Node zu verstehen, welcher mehrere Nodes zusammenfasst. Somit kann ein Prozess noch detaillierter dargestellt werden. Ein Metanode besitzt Input- und Outputports, welche nach belieben angepasst werden können.

- Component

Ein Component Node fasst wie ein Metanode mehrere Nodes zusammen. Im Gegensatz zum Metanode besitzt ein Component Node jedoch keine Ports. Ein Component Node hat die Besonderheit, dass die Nodes welche in der Component enthalten sind, komplett getrennt vom Rest des Workflows sind. Dies bedeutet, dass ausserhalb des Component Nodes nicht auf den Inhalt zugegriffen werden kann.

7.2 Camunda Modeler

7.2.1 Properties Panel Hinzufügen

Das Properties Panel ist ein Panel, welches als eigenes Modul in die Applikation eingebunden werden kann. Das Properties Panel Modul sowie der Provider müssen importiert und dem BpmnModeler der Applikation hinzugefügt werden.

```
import BpmnModeler from 'bpmn-js/lib/Modeler';
import {
  BpmnPropertiesPanelModule,
  BpmnPropertiesProviderModule,
} from '@bpmn-io/bpmn-properties-panel';

const modeler = new BpmnModeler({
  container: '#canvas',
  propertiesPanel: {
    parent: '#properties'
  },
  additionalModules: [
    BpmnPropertiesPanelModule,
    BpmnPropertiesProviderModule
  ]
});
```

Properties zum Properties Panel hinzufügen

Wenn man das Properties Panel um eigene Properties erweitern möchte, muss dafür ein eigener PropertiesPanelProvider geschrieben werden. Dieser wird dann mithilfe von Injection dem PropertiesPanel hinzugefügt.

In diesem Provider kann bei den Gruppen im Provider Elemente hinzufügen, ändern oder entfernen.

```
class ExamplePropertiesProvider {
  constructor(propertiesPanel) {
    propertiesPanel.registerProvider(500, this);
  }

  getGroups(element) {
    return (groups) => {

      // add, modify or remove groups
      // ...

      return groups;
    };
  }
}
```

```
ExamplePropertiesProvider.$inject = [ 'propertiesPanel' ];
```

7.2.2 Palette anpassen

Um die Palette von Camunda Modeler anzupassen, müssen folgende Elemente erstellt werden:

Vorbereitung

Zur Erstellung von Custom-Elementen werden mehrere Files benötigt, es macht darum Sinn einen Ordner anzulegen um alle relevanten Files darin abzulegen. Für dieses Beispiel haben wir einen Ordner namens "custom" im Editor erzeugt. Dieser ist im Pfad "src/app/tabs/bpmn" zu finden wo ebenfalls der Rest des Editors implementiert ist.

Beschreibung der Komponenten

Der PaletteProvider ist dazu da eine Palette zur Verfügung zu stellen. In diesem können bestehende Elemente angepasst oder neue hinzugefügt werden. Seine Funktion besteht nur darin die Elemente aufzulisten.

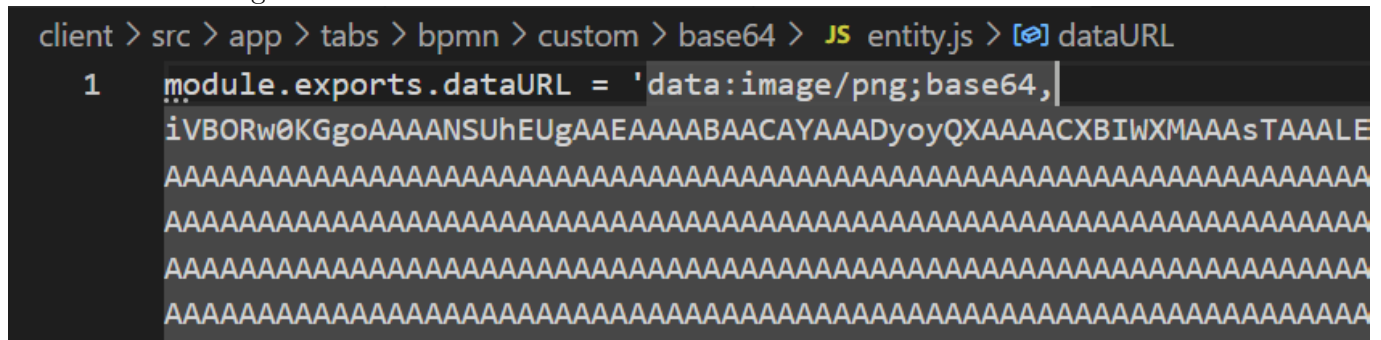
Der Renderer ist dazu da das Element zu zeichnen wenn es in die Zeichenmappe gezogen wird. In diesem wird definiert wie das Element gezeichnet wird.

Bild zur Darstellung eines Elements

Das Bild welches verwendet werden soll ist im besten Fall ein SVG, da sich dieses sauber skaliert und nicht verpixelt wenn es in der Zeichenfläche gerendert wird. Damit das Bild anschliessend einfach eingebettet werden kann empfehlen wir dieses mittels base64 zu encoden und als Javascript-Dokument abzulegen. Die encodierung kann beispielsweise mit einem online encoder erfolgen. Der dazu generierte base64 String kann dann folgendermassen im Javascript eingesetzt werden.

```
module.exports.dataURL = 'SVG base64 String';
```

Dies könnte wie folgt aussehen:



```
client > src > app > tabs > bpmn > custom > base64 > JS entity.js > dataURL
1  module.exports.dataURL = 'data:image/png;base64,
  iVBORw0KGgoAAAANSUhEUgAAEAAAABAACAYAAADyoyQXAAAACXBIWXMAAAAsTAALE
  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
  AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA
```

Bild 7.1: Javascript mit base64 SVG Bild

Palette Provider erstellen

```
import Entity from '../base64/entity' // item to be added to the palette aka. your SVG image

/**
 * A provider for quick service task production
 */
export default function PaletteProvider(palette, create, elementFactory) {

  this._create = create;
  this._elementFactory = elementFactory;

  palette.registerProvider(this);
}

PaletteProvider.$inject = [
  'palette',
  'create',
  'elementFactory'
];

PaletteProvider.prototype.getPaletteEntries = function() {

  var elementFactory = this._elementFactory,
  create = this._create;

  function startCreate(event) {
    var serviceTaskShape = elementFactory.create(
```

```

    'shape', { type: 'bpmn:ServiceTask' }
  );

  create.start(event, serviceTaskShape);
}

return {
  'create-Entity': {
    group: 'activity',
    title: 'Create a new Entity',
    imageUrl: Entity.dataURL,
    action: {
      dragstart: startCreate,
      click: startCreate
    }
  }
};
};
};

```

Custom Renderer implementieren

```

import inherits from 'inherits';
import BaseRenderer from 'diagram-js/lib/draw/BaseRenderer';

import {
  is
} from 'bpmn-js/lib/util/ModelUtil';

import Entity from '../base64/entity'

import {
  append as svgAppend,
  create as svgCreate
} from 'tiny-svg';

//examples:
//https://github.com/bpmn-io/bpmn-js-example-custom-shapes
//https://github.com/bpmn-io/bpmn-js-examples/tree/master/custom-elements
//https://bpmn.io/toolkit/bpmn-js/

export default function Render(eventBus) {
  BaseRenderer.call(this, eventBus, 1500);

  this.canRender = function(element) {
    return is(element, 'bpmn:ServiceTask');
  };

  this.drawShape = function(parent, shape) {

    var asdf = shape.labels[0];
    var url = '';

    switch(shape.type)
    {
      case "bpmn:" : url = Process.dataURL;

      default: url = Entity.dataURL;
    }

    var entityGfx = svgCreate('image', {

```

```
        x: 0,
        y: 0,
        width: shape.width,
        height: shape.height,
        href: url
    });

    svgAppend(parent, entityGfx);

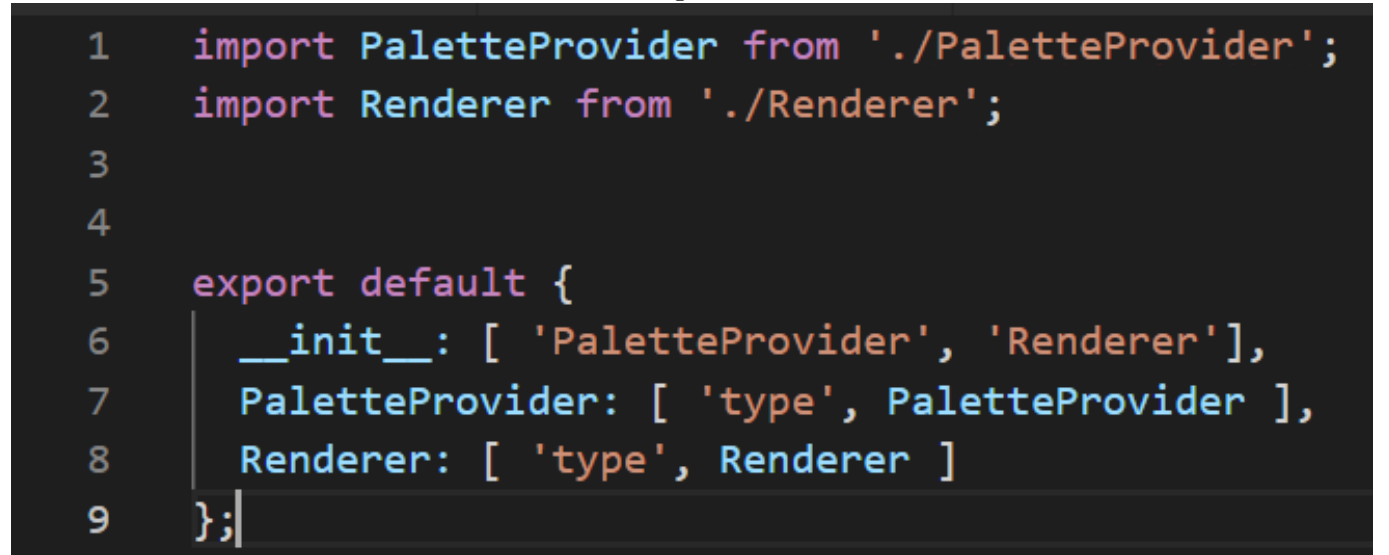
    return entityGfx;
  };
}

inherits(Render, BaseRender);

Render.$inject = [ 'eventBus' ];
```

Custom Elemente in Camunda Modeler einfügen

Um die erstellen Files einfach zu integrieren kann man ein Index.js File erstellen worin der Palette-Provider sowie der Renderer definiert ist. Dies könnte wie folgt aussehen:



```
1  import PaletteProvider from './PaletteProvider';
2  import Renderer from './Renderer';
3
4
5  export default {
6    __init__: [ 'PaletteProvider', 'Renderer' ],
7    PaletteProvider: [ 'type', PaletteProvider ],
8    Renderer: [ 'type', Renderer ]
9  };
```

Bild 7.2: Index.js

Zuletzt muss noch der BpmnEditor selbst bearbeitet werden. Das erstellte Modul kann folgendermassen im Editor hinzugefügt werden.

```
import customModule from './custom';

const modeler = new BpmnModeler({
  ...options,
  position: 'absolute',
  additionalModules: [
    customModule
  ],
});
```

Kapitel 8

Referenzen

- [1] Awesome Open Source, *Liste von open source bpmn projekten*, <https://awesomeopensource.com/projects/bpmn>.
- [2] KNIME AG, Zurich, Switzerland, *Knime sdk*, <https://github.com/knime/knime-sdk-setup/>.

Kapitel 9

Eigenständigkeitserklärung

Hiermit erklären wir, dass es sich bei der von uns eingereichten schriftlichen Arbeit mit dem Titel **simBPMN Visualizer** um eine Originalarbeit handelt, die wir allein verfasst haben und die in unseren eigenen Worten verfasst ist.

Autoren¹

Michel

Mirsayyah

Sven

Höpfner

Betreuender Dozent

Mit der Unterschrift erklären wir, dass wir über die normalen akademischen Zitierregeln informiert worden sind und die Informationen zur Zitieretikette² gelesen und verstanden haben. Die üblichen Zitierkonventionen wurden eingehalten. Die oben genannte schriftliche Arbeit kann elektronisch auf Plagiate geprüft werden.

Rapperswil 24.12.2021

Ort,

Datum,

Unterschrift (Michel Mirsayyah)

Rapperswil 24.12.2021

Ort,

Datum,

Unterschrift (Sven Höpfner)

¹Mitverfasste Arbeit: Die Unterschriften aller Autoren sind erforderlich. Jede Unterschrift bestätigt die Originalität des gesamten schriftlichen Werkes in seiner endgültigen Form

²<https://www.ethz.ch/content/dam/ethz/main/education/rechtliches-abschluesse/leistungskontrollen/plagiarism-citationetikette.pdf>

Kapitel 10

Glossar

BPMN 2.0

Business Process Model and Notation.

Eclipse

Eclipse ist eine Standard [IDE](#) welche in der Softwareentwicklung eingesetzt wird..

Java

Java ist eine typsichere, objektorientierte Allzweck-Programmiersprache.

JSON

Die JavaScript Object Notation (JSON) ist ein kompaktes Datenformat in einer einfach lesbaren Textform für den Datenaustausch zwischen Anwendungen. JSON ist von der Programmiersprache unabhängig.

KNIME

KNIME ist eine auf [Eclipse](#) basierende Umgebung, in welcher Workflows abgebildet werden können..

MVP

MVP steht für Minimum Viable Product, was die minimal brauchbare Applikation darstellt.

Node

Ein Node ist ein allgemeines Objekt in einem Workflow..

Nuget

NuGet ist ein System zur freien Verteilung von Software-Komponenten in Form von Paketen in Microsoft .Net Projekten..

Object Management Group

Die OMG ist eine internationale not-for-Profit Organisation welche Technologie-Standards definiert..

OST

Ostschweizer Fachhochschule.

Scrum

Scrum ist ein Vorgehensmodell des Projekt- und Produktmanagements, insbesondere zur agilen Softwareentwicklung. Es wurde ursprünglich in der Softwaretechnik entwickelt, ist aber davon unabhängig.

Scrum+

Eine Vermischung des Wasserfallmodells und Scrum. Gearbeitet wird nach Scrum, ergänzt um Meilensteine aus dem Wasserfallmodell.

simBPMN

Erweiterte Business Process Model and Notation.

UML

Unified Modeling Language.

WPF

Windows Presentation Foundation, auch bekannt unter dem Codenamen Avalon, ist ein Grafik-Framework und Fenstersystem des .NET Frameworks von Microsoft ([siehe Wikipedia](#)).

XML

Die Extensible Markup Language, abgekürzt XML, ist eine Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten..

YouTrack

Issue Tracker- sowie Zeiterfassungstool von JetBrains.

Kapitel A

Sitzungsprotokolle

A.1 Kick-Off Meeting

Woche 1: 20. September 2021, 17:30 – 18:30

Sitzungsteilnehmer

- Prof. Dr. Andreas Rinkel
- Michel Mirsayyah
- Sven Höpfner

Traktanden

- Kick-Off SA
- Vorstellung Thema Rinkel
- Festlegung des Themas

Beschlüsse (Diskussion)

- Das Team wird das Thema [simBPMN](#) umsetzen

Offene Punkte (erledigt vor nächster Sitzung)

Was	Verantwortlichkeit
Einarbeitung in das Thema simBPMN	Michel, Sven

A.2 SimBPMN Workshop Meeting

Woche 2: 24. September 2021, 13:00 – 15:00

Sitzungsteilnehmer

- Prof. Dr. Andreas Rinkel
- Michel Mirsayyah
- Sven Höpfner

Traktanden

- Workshop Thema [simBPMN](#)
- Abklärung Vorgehen Studienarbeit

Beschlüsse (Diskussion)

- Projektplan erstellen
- Zwischenabgabe nach Bachelorarbeit (unbewertet)
- Bei Abschlusspräsentation Abfragung nach Bachelorarbeit (unbewertet)
- Abgabe von Schreibproben an Herrn Rinkel während des Projektes
- Projektplan erstellen

Offene Punkte (erledigt vor nächster Sitzung)

Was	Verantwortlichkeit
Abklärung von Technologien für die Umsetzung	Michel, Sven

A.3 Meeting zum Meilenstein 1 (Projektplan)

Woche 3: 07. Oktober 2021, 16:30 – 17:30

Sitzungsteilnehmer

- Prof. Dr. Andreas Rinkel
- Michel Mirsayyah
- Sven Höpfner

Traktanden

- Besprechung gefundene Technologien
- Besprechung Projektplan

Beschlüsse (Diskussion)

- Zwischenabgabe der Dokumentation nach dem Meilenstein "End of Elaboration".
- Die Applikation soll nur dann selbst, wenn es keine andere Möglichkeit gibt. Am besten man arbeitet auf einem bereits vorhandenem Framework.

Offene Punkte (erledigt vor nächster Sitzung)

Was	Verantwortlichkeit
Evaluieren ob man mithilfe von Python die Applikation umsetzen kann	Michel
Evaluieren in wie fern KNIME verwendet werden kann.	Sven

A.4 Vorstellungsmeeeting KNIME

Woche 4: 14. Oktober 2021, 16:30 – 17:30

Sitzungsteilnehmer

- Prof. Dr. Andreas Rinkel
- Michel Mirsayyah
- Sven Höpfner

Traktanden

- Vorstellung KNIME
- Vorstellung Python Funde

Beschlüsse (Diskussion)

- Verwendung von KNIME für die Arbeit.

Offene Punkte (erledigt vor nächster Sitzung)

Was	Verantwortlichkeit
Erstellung eines KNIME Projekts mit einfachen Nodes	Sven
Erstellung der Anwendungsfälle für ein MVP	Michel

A.5 Knime und simBPMN

Woche 5: 21. Oktober 2021, 16:30 – 17:30

Sitzungsteilnehmer

- Prof. Dr. Andreas Rinkel
- Michel Mirsayyah
- Sven Höpfner

Traktanden

- Besprechung Knime mit simBPMN
- Besprechung der ersten Anwendungsfälle

Beschlüsse (Diskussion)

- KNIME wird nicht für das verwendet fürs was es gedacht ist. Allerdings wird versucht, wie gut man simBPMN in KNIME einbauen kann.
- simBPMN ist eigentlich dafür gedacht einfach verwendet werden zu können. KNIME könnte da zu komplex werden. Es soll untersucht werden, ob dies der Fall ist.
- Des weiteren soll untersucht werden, wie man allfällige Schwachstellen/fehlende Funktionen implementieren beziehungsweise ob man einen Workaround finden kann.
- Herr Knut Schmahl übernimmt die Rolle des Experten in der Arbeit

Offene Punkte (erledigt vor nächster Sitzung)

Was	Verantwortlichkeit
KNIME auf fehlende Funktionen überprüfen und dafür eine Lösung / einen Workaround finden	Sven, Michel

A.6 Vorstellungsmeeing KNIME

Woche 6: 28. Oktober 2021, 16:30 – 17:30

Sitzungsteilnehmer

- Prof. Dr. Andreas Rinkel
- Michel Mirsayyah
- Sven Höpfner

Traktanden

- Besprechung von KNIME im Zusammenhang mit simBPMN
- Besprechung weiteres Vorgehen

Beschlüsse (Diskussion)

- KNIME wird nicht weiter in der Arbeit verfolgt. Es wird Dokumentiert, wieso KNIME vielversprechend aussah, es jedoch nicht in der Zielgruppe entspricht.
- Es wurde beschlossen, nach 2-3 weiteren Optionen zu suchen und diese Auszuwerten.
- In der nächste Woche wird beschlossen, welches Projekt weiter angeschaut wird, in welchem dann auch ein Durchstich sowie 2-3 User Stories erstellt werden.

Offene Punkte (erledigt vor nächster Sitzung)

Was	Verantwortlichkeit
Suchen und Auswerten von weiteren Option zur Darstellung von Workflows	Sven, Michel

A.7 Vorstellungsmeeting Alternative

Woche 7: 04. November 2021, 16:30 – 17:30

Sitzungsteilnehmer

- Prof. Dr. Andreas Rinkel
- Michel Mirsayyah
- Sven Höpfner

Traktanden

- Besprechung der neuen Funde zur Darstellung von Workflows
- Besprechung weiteres Vorgehen

Beschlüsse (Diskussion)

- BPMN-JS und die Grundlegende Architektur dazu sieht sehr vielversprechend aus und wird weiter Analysiert.
- Die Dokumentation wird nun aufgearbeitet und die bisherige Arbeit beschrieben
- Eine Dokumentation soll für das neue System erstellt werden (Wie ist man dazu gekommen, was soll der Prototyp können und was muss noch verändert werden.

Offene Punkte (erledigt vor nächster Sitzung)

Was	Verantwortlichkeit
Dokumentation aufarbeiten	Sven, Michel

A.8 Vorstellungsmeeing Camunda Modeler & BPMN Studio

Woche 8: 11. November 2021, 16:30 – 17:30

Sitzungsteilnehmer

- Prof. Dr. Andreas Rinkel
- Michel Mirsayyah
- Sven Höpfner

Traktanden

- Vorstellung Camunda Modeler & BPMN Studio und deren Unterschiede
- Besprechung weiteres Vorgehen

Beschlüsse (Diskussion)

- Aufgrund der vielen Abhängigkeiten wird BPMN Studio nicht weiter verfolgt und der Camunda Modeler näher angeschaut.
- Es wird untersucht in wie fern der Camunda Modeler erweiterbar ist

Offene Punkte (erledigt vor nächster Sitzung)

Was	Verantwortlichkeit
Erstellung eines ersten Prototyps für den Camunda Modeler	Sven, Michel
Erweiterbarkeit untersuchen und entsprechend einbauen	Sven, Michel

A.9 Fortschritt Camunda Modeler

Woche 9: 18. November 2021, 16:30 – 17:30

Sitzungsteilnehmer

- Prof. Dr. Andreas Rinkel
- Michel Mirsayyah
- Sven Höpfner

Traktanden

- Besprechung der Erkenntnisse beim Camunda Modeler
- Besprechung weiteres Vorgehen

Beschlüsse (Diskussion)

- Da der Camunda Modeler vielversprechend aussieht, soll dieser noch für die nächste Woche weiter untersucht werden.
- Es wird ein Termin mit dem Herrn Knut Schmahl für die Zwischenpräsentation gesucht.

Offene Punkte (erledigt vor nächster Sitzung)

Was	Verantwortlichkeit
Prototyp des Camunda Modelers ausarbeiten	Sven, Michel
Termin mit Knut Schmahl für die Zwischenpräsentation finden	Sven

A.10 Vorbereitung Zwischenpräsentation

Woche 10: 25. November 2021, 16:30 – 17:30

Sitzungsteilnehmer

- Prof. Dr. Andreas Rinkel
- Michel Mirsayyah
- Sven Höpfner

Traktanden

- Besprechung Fortschritt Entwicklung des Prototyps
- Besprechung Durchführung der Zwischenpräsentation

Beschlüsse (Diskussion)

- Der derzeitige Prototyp wird abgeschlossen und nicht weiter untersucht.
- Die Zwischenpräsentation wird im Verlauf der nächsten Woche abgehalten.

Offene Punkte (erledigt vor nächster Sitzung)

Was	Verantwortlichkeit
Die Zwischenpräsentation vorbereiten und abhalten	Sven, Michel

A.11 Feedback Zwischenpräsentation

Woche 11: 02. Dezember 2021, 16:30 – 17:00

Sitzungsteilnehmer

- Prof. Dr. Andreas Rinkel
- Michel Mirsayyah
- Sven Höpfner

Traktanden

- Besprechung der Zwischenpräsentation
- Besprechung über die Möglichkeit BPMN IO in eine frische Electron Applikation einzubauen

Beschlüsse (Diskussion)

- Die Zwischenpräsentation ist sehr gut angekommen, mit wenigen negativen Punkten.
- Bis nächste Woche wird versucht eine frische Electron Applikation aufzusetzen und BPMN IO darin einzubauen.
- Die Dokumentation wird ab nächster Woche jeweils wöchentlich für Zwischenabgaben abgegeben.

Offene Punkte (erledigt vor nächster Sitzung)

Was	Verantwortlichkeit
Ein Prototyp einer eigenen Electron Applikation aufzusetzen	Sven
Dokumentation für eine Zwischenabgabe vorbereiten	Michel

A.12 Abschluss Entwicklung

Woche 12: 09. Dezember 2021, 16:30 – 17:00

Sitzungsteilnehmer

- Prof. Dr. Andreas Rinkel
- Michel Mirsayyah
- Sven Höpfner

Traktanden

- Besprechung Erkenntnisse von eigener BPMN Applikation
- Besprechung weiteres Vorgehen

Beschlüsse (Diskussion)

- Jegliche Entwicklungsarbeit wird nun eingestellt und es wird nur noch an der Dokumentation gearbeitet.

Offene Punkte (erledigt vor nächster Sitzung)

Was	Verantwortlichkeit
Feedback zur Dokumentation einbauen	Sven, Michel

A.13 Abschluss Studienarbeit

Woche 13: 16. Dezember 2021, 16:30 – 17:00

Sitzungsteilnehmer

- Prof. Dr. Andreas Rinkel
- Michel Mirsayyah
- Sven Höpfner

Traktanden

- Besprechung des Status der Dokumentation
- Klärung allfälliger Fragen

Beschlüsse (Diskussion)

- Für die Abgabe der Arbeit wird keine physische Form verlangt.
- Die letzte Sitzung am 23.12.2021 fällt aus.

Offene Punkte (erledigt vor nächster Sitzung)