

**Universität Leipzig**  
Wirtschaftswissenschaftliche Fakultät  
Institut für Wirtschaftsinformatik  
Professur Anwendungssysteme

**Dokumentation**

im Rahmen der Veranstaltung „Geschäftsprozessmanagement in der Finanzindustrie“ im  
Masterstudiengang Wirtschaftsinformatik

**Recherche zur Ermittlung von Berechtigten für Finanzhilfen**

Betreuer im Unternehmen: Dr. Axel Hoppe  
Axel.Hoppe@commerzbank.de

Betreuender Hochschullehrer: Prof. Dr. Rainer Alt  
Betreuender Assistent: Christian Dietzmann  
Bearbeiter: Felix Brockschnieder  
Oleg Chapaykin  
Ravel Siirde  
Benedikt Kaiser  
Florian Göhring  
Eingereicht am: Dienstag, 1. Juni 2021

## Gliederung

<b>Gliederung.....</b>	<b>I</b>
<b>Abbildungsverzeichnis .....</b>	<b>II</b>
<b>Tabellenverzeichnis .....</b>	<b>Fehler! Textmarke nicht definiert.</b>
<b>Abkürzungsverzeichnis .....</b>	<b>III</b>
<b>1 Beschreibung und Problemstellung .....</b>	<b>1</b>
1.1 Case-Beschreibung .....	1
1.2 Dependencies .....	1
<b>2 Aufgabenverteilung und Projektorganisation .....</b>	<b>2</b>
<b>3 Lösungsprozess .....</b>	<b>4</b>
3.1 Aufgabenteilung .....	4
3.2 Schnittstellenfestlegung .....	4
3.3 Bereichslösungen .....	5
3.3.1 Eingabe .....	5
3.3.2 Verarbeitung .....	10
3.3.3 Ausgabe .....	12
3.4 Zusammenführung .....	13
3.5 Fragen und Probleme .....	13
3.5.1 Zusammenarbeit .....	13
3.5.2 Company ID .....	14
3.5.3 Definierung von Schnittstellen .....	14
<b>4 Auswirkungen auf den Referenzprozess .....</b>	<b>15</b>
4.1 Prozessveränderung .....	15
4.2 Wertbeitrag .....	16
4.3 Strategische Bedeutung .....	18
<b>Literaturverzeichnis .....</b>	<b>Fehler! Textmarke nicht definiert.</b>

**Abbildungsverzeichnis**

Abbildung 1 - Prozess "Abgleich mit Bestandskunden" .....	10
Abbildung 2 - Überprüfung der Collection .....	11
Abbildung 3 - Referenzprozess .....	15

## Abkürzungsverzeichnis

CSV	Comma-separated values, Dateiformat zur Speicherung strukturierter Daten
XML	Extensible Markup Language, Auszeichnungssprache zur Darstellung hierarchisch strukturierter Daten
MVP	Minimum Viable Product, Erste minimal funktionsfähige Iteration eines Produkts
PLZ	Postleitzahl
ID	Identifikationsnummer
POP/POP3	Post Office Protocol (3), Übertragungsprotokoll, über das ein Client E-Mails von einem E-Mail-Server abholen kann
SMTP	Simple Mail Transfer Protocol, Protokoll zum Austausch von E-Mails in Computernetzen
KfW	Förderbank
IT	Informationstechnik
IfM Bonn	Institut für Mittelstandsforschung Bonn
KMU	Kleine und mittlere Unternehmen, Unternehmen, die definierte Grenzen hinsichtlich Beschäftigtenzahl, Umsatzerlös oder Bilanzsumme nicht überschreiten

# 1 Beschreibung und Problemstellung

Im Rahmen der Veranstaltung „Geschäftsprozessmanagement 2“ stellte die Partnerfirma ComTS eine Automatisierungsaufgabe, um die gelernten theoretischen Inhalte des Moduls zu verfestigen. In dieser Aufgabe sollte eine Gruppe aus ca. fünf Studierenden gemeinsam einen Prozess innerhalb der Software „Blue Prism“ automatisieren, diesen technisch und formal dokumentieren und eine Präsentation vorbereiten.

## 1.1 Case-Beschreibung

Der Case „Recherche zur Ermittlung von Berechtigten für Finanzhilfen“ beschreibt die automatische Suche von potenziellen Unternehmen, welche eine Corona-Hilfe an Anspruch nehmen können, und die Vorbereitung von Informationsbriefen. Für den Prozess wird eine Mail mit einem Anhang bereitgestellt. Im Anhang befindet sich eine CSV-Tabelle, deren Inhalt aus Kombinationen von Branchen und Postleitzahlen bildet. Anhand dieser Postleitzahlen sollen automatisch die entsprechenden Unternehmen aus <https://www.dastelefonbuch.de/> herausgesucht und zwischengespeichert werden. Um zu verhindern, dass Unternehmen, die bereits einen Antrag auf Finanzhilfe gestellt haben, erneut angeschrieben werden, wird ein Abgleich der zwischengespeicherten Unternehmen mit einer Liste von Kundendaten abgeglichen. Die bereinigte Liste mit den zu informierenden Unternehmen wird im letzten Schritt weiterverarbeitet, in dem eine Word-Vorlage mit den relevanten Informationen befüllt wird. Diese Dateien werden in einer .zip-Datei an eine festgelegte E-Mail-Adresse gesendet. Damit ist der Prozess abgeschlossen.

## 1.2 Dependencies

Externe Business Objekte werden unter `utils/` bereitgestellt. Quellen:

- <https://github.com/blue-prism/pop3-smtp-utility>
- <https://github.com/blue-prism/file-management-utility>
- <https://github.com/blue-prism/collection-manipulation>
- <https://github.com/blue-prism/strings-utility.git>
- [https://digitalexchange.blueprism.com/dx/entry/113833/solution/atcs---compression-and-extraction?\\_ga=2.164223996.275969242.1622196232-1643744417.1621856680](https://digitalexchange.blueprism.com/dx/entry/113833/solution/atcs---compression-and-extraction?_ga=2.164223996.275969242.1622196232-1643744417.1621856680)

## 2 Aufgabenverteilung und Projektorganisation

Das Projekt wurde von den Studierenden selbstständig in drei wesentliche Bestandteile gegliedert: Eingabe, Verarbeitung, Ausgabe. Die Dokumentation des Projektes wurde untereinander in die entsprechenden Kompetenzbereiche aufgeteilt, um die präziseste fachliche Beschreibung zu gewährleisten.

Die Eingabe der Daten, bearbeitet von Felix Brockschnieder und Benedikt Kaiser, umfasste die Tätigkeiten:

- E-Mail-Empfang
- Auslesen des Anhangs
- Verarbeitung jeder Zeile der CSV:
  - o Aufruf der Internetseite mit entsprechenden Parametern
  - o Auslesen der Ergebnisse
  - o Speicherung der Unternehmen in einer Queue
- Fertigstellen des Eingabeprozesses
- Fehlerhandling

Folgende Tätigkeiten sind für die Bearbeitung des Datenabgleichs durch Florian Göhring getätigt worden:

- Lesen der Elemente aus der Queue
- Abgleich des Queue-Elements mit der vorgegebenen Collection
- Auswahl des Elementes, falls es nicht in der Collection vorkommt
- Erstellung einer neuen Queue für die zu informierenden Unternehmen
- Zusammenführung der Teilprozesse in einen Gesamtprozess
  - o Kompatibilität zwischen den Prozessen
  - o Debugging
  - o Refactor
  - o Test

Um den identifizierten zu informierenden Unternehmen ein adäquates Anschreiben zu generieren, übernahmen Oleg Chapaykin und Ravel Siirde die Aufgaben der Ausgabe:

- Lesen der Elemente aus der Queue
- Eingabe-Algorithmus in der Word-Vorlage
- Erstellung des Dokumentenarchivs
- Senden des Archivs an die definierte Mailadresse

Am Ende der individuellen Bearbeitung der Cases wurden die Teilprozesse zusammengefügt, um eine einzelne Ausführung des automatisierten Prozesses zu ermöglichen.

Die Kommunikation im Projekt fand primär über das Tool „Discord“ statt. Für die Konzepterstellung, Fragedokumentation und Zeichnungen diente „Microsoft OneNote“ als kollaboratives Notizen-Tool.

### 3 Lösungsprozess

Um die vorliegende Lösung zu entwickeln, hat das Team verschiedene Tätigkeiten durchgeführt, um eine effiziente Zusammenarbeit zu erreichen. Im Folgenden werden diese Aufgaben beschrieben.

#### 3.1 Aufgabenteilung

Um parallel an dem Projekt arbeiten zu können, wurden die verschiedenen Bereiche des Prozesses - Eingabe, Verarbeitung, Ausgabe, aufgeteilt auf die Studierenden. Dies ermöglichte eine zeitsparende, parallele Entwicklung. Die Voraussetzung war an der Stelle das gemeinsame Verständnis der Architektur sowie die klare Abgrenzung der durchzuführenden Prozessschritte. Da es zu Beginn einige Fragen bzgl. der Zusammenarbeit gab, wurde die Möglichkeit einer technischen Unterstützung seitens Herrn Hoppe in Anspruch genommen, der die bestehenden Fragen klären und somit die Voraussetzungen der Zusammenarbeit geklärt werden konnte. Da BluePrism keine Möglichkeiten für das kollaborative Arbeiten an einem Projekt bietet entschieden wir uns für das Anlegen eines GitHub-Repositories. In dieses wurden die in BluePrism erstellten Business-Objekte und Prozesse über die Export-Funktion gespeichert. Da BluePrism diese in Textfiles mit XML-ähnlicher Struktur speichert konnten so die Vorteile der Versionsverwaltung genutzt werden und der aktuelle Stand des Projekts war stets für alle Studierenden verfügbar.

#### 3.2 Schnittstellenfestlegung

Um die einzelnen Teilbereiche miteinander zu verbinden, waren zwei Entscheidungen zu treffen:

1. Wie werden die einzelnen Prozess-Abschnitte entwickelt, um sie am Ende einfach und effizient zusammenführen zu können?
2. Welche Daten werden auf welchem Weg zwischen den Prozessschritten miteinander ausgetauscht? Die technische Hilfestellung von Herrn Hoppe führte auch hier dazu, dass eine effiziente Entscheidung getroffen werden konnte. Die Prozessschritte wurden in BluePrism als unabhängige Prozesse entwickelt, was es ermöglicht diese später von einem Hauptprozess aus als Unterprozesse aufzurufen.



Mit der Entscheidung, die zu bearbeitenden Unternehmen über Warteschlangen (Queues) zwischen den Prozessschritten auszutauschen, wurde die unabhängige Entwicklung der Teilbereiche ermöglicht. Im Anschluss konnten die Studierenden mit der Arbeit beginnen und die Bereichslösungen implementieren.

### **3.3 Bereichslösungen**

Die parallele Entwicklung an Eingabe, Verarbeitung und Ausgabe ist bei dem gesteckten Zeitfenster essenziell für den Erfolg des Projekts. So können die Teammitglieder gleichzeitig an den Lösungen arbeiten. Das gemeinschaftliche Ziel war die unabhängige Entwicklung, um schnellstmöglich eine funktionierendes Minimum Viable Product (MVP) vorliegen zu haben, welches dann im weiteren Verlauf ergänzt und optimiert werden kann.

#### **3.3.1 Eingabe**

Der erste Prozessschritt besteht im Wesentlichen aus zwei Abschnitten. Zunächst müssen die per E-Mail versendeten Input-Datensätze mit Branche und Postleitzahl (PLZ) entgegengenommen werden und innerhalb BluePrisms als Collection abgelegt werden. Im Anschluss wird für jeden Datensatz das Telefonbuch durchsucht um eine Liste aller zu bearbeitenden Unternehmen und deren Adressen zu erhalten. Diese werden dann für den nächsten Prozessschritt in eine Queue übergeben.

#### **Input**

Als Input für den Prozess wird eine E-Mail von einem Mitarbeiter an [blueprism.cobitest@gmail.com](mailto:blueprism.cobitest@gmail.com) gesendet. Im Anhang müssen sich eine oder mehrere CSV-Dateien befinden. Damit die Daten später korrekt ausgelesen werden können, ist es wichtig, dass die Datei eine bestimmte Struktur hat. So repräsentiert die erste Spalte die Postleitzahl, während in der zweiten Spalte die Branche steht.

#### **Output**

Nachdem der komplette Eingabe Prozess durchgelaufen ist, werden die gefundenen Unternehmen in die Queue übergeben. Jeder Eintrag, der in die Queue gegeben wird, hat dieselbe Struktur. Er besteht aus einer einzigartigen Company ID, dem Name, der Straße und Hausnummer, Postleitzahl, Stadt und Branche.

## Voraussetzungen

Damit der Prozess flüssig läuft, müssen mehrere externe Objekte (Pop3/SMTP, strings, file management und collection manipulation) heruntergeladen und in Blue Prism importiert werden. Des Weiteren muss auf der Main Page ein lokaler Pfad festgelegt werden, auf dem die CSV-Dateien gespeichert werden.

## Prozess

### Main Page:

Die Main Page dient als Prozesshülle für die beiden wichtigen Subprozesse Query input und Process message. Process message wird dabei in einer Schleife für jede message ausgeführt.

### Query Input:

- Konfiguration mit Hilfe des Pop3/SMTP business object und Auflistung der messages in einer collection
- für jede message:
  - o Generierung der messageID und Erstellung eines subfolders unter dem vorher angegebenen lokalen Pfad
  - o Speichern des Anhangs in diesem subfolder und danach löschen der E-Mail (Pop3/SMTP)

### Process message:

Inputs: messagePath und messageID

#### 1. Read from CSV Input: CSV Path

- a. mit dem file management business object werden die Dateien des subfolders in eine collection importiert
- b. für jede Datei in dieser collection: - file management bo: die CSV-Datei als Text in die Input Data collection importieren - über jede Zeile der Input Data collection iterieren und die Postleitzahl kontrollieren. Je nach Formatierung kann es passieren, dass die Nullen am Anfang der PLZ weggelassen werden, was zu Problemen bei der Suche führt. Daher wird in diesem Schritt kontrolliert, ob die PLZ 5 Ziffern hat und wenn nicht, werden an den Anfang entsprechend viele Nullen angehängt. - anhängen der Zeilen an die collection  
Jobs Output: Query Jobs collection mit den Werten aus der Jobs collection

2. iterieren über die Query Jobs Collection und damit über jedes PLZ/Branchen Paar
  - a. die unten beschriebenen Aktionen des Business Objects **Das Telefonbuch** werden nacheinander aufgerufen und der Output davon an die Collection ResultsTotal angehängt.
3. Results to Queue Input: ResultsTotal collection als Data
  - a. Generierung der CompanyID: für jede Firma wird aus der Branche, der PLZ, dem Name und der Straße eine eindeutige CompanyID generiert
  - b. die Companys collection wird in die Queue übergeben

Damit ist der Eingabe-Prozess beendet und der Verarbeitung-Prozess kann mit den Daten in der Queue starten.

### **Das Telefonbuch - Business Object**

Für jeden Eintrag bestehend aus Branche und PLZ wird anschließend die Webseite [www.dastelefonbuch.de](http://www.dastelefonbuch.de) nach entsprechenden Unternehmen durchsucht. Für die Interaktion zwischen Blue Prism und der Webseite wurde ein Business Object angelegt, das als *Browser-based Application* konfiguriert ist. Als Browser wurde Google Chrome gewählt, da dieser mit den integrierten Entwickler-Werkzeugen ein effizientes Debugging bei der Definition von HTML-Elementen für das *Application-Model* ermöglicht.

#### **i) Application-Model**

Das Application-Model besteht aus zwei primären Elementen, der "Startseite" und den Ergebnissen ("Results").

Die Startseite ist die Webseite, die bei Aufruf der URL [www.dastelefonbuch.de](http://www.dastelefonbuch.de) zurückgegeben wird. Innerhalb dieses Elements sind als Kind-Elemente die zwei Suchfelder "Branche" und "PLZ" definiert, sowie der Button "Finden". Die drei Kind-Elemente werden statisch durch ihren *Web Path* referenziert, da nicht davon auszugehen ist, dass sich das primäre Interface der Startseite häufig ändert.

Die Seite mit den Suchergebnissen besteht zum einen aus einem Element namens "Entry", das die Felder mit den gesuchten Information als Unterelemente definiert und zum anderen aus Elementen die zur Unterstützung des Web-Crawlings benötigt werden.

Das "Entry"-Element hat fünf Unterelemente:

- Name
- Straße
- PLZ
- Ort
- Stadtteil

Da für eine Anfrage idR. mehrere Ergebnisse zurückgibt, werden diese jeweils über mittels einer dynamischen XPath-Query identifiziert. Das bedeutet, dass der jeweilige Ausdruck zur Laufzeit des Prozesses zusammengesetzt wird, was innerhalb des Prozesses "Crawl-Information" unten näher beschrieben ist. Zur Unterstützung des Crawlings definierte Elemente sind:

- Total Results - die Anzahl an zurückgegeben Ergebnissen pro Anfrage.
- Keine Treffer Nachricht – Benachrichtigung, die erscheint, wenn keine Ergebnisse für die gesuchte PLZ gefunden wurden und stattdessen Ergebnisse für benachbarte PLZ zurückgegeben werden.
- Footer - Das Footer Element am unteren Ende der Webseite (benötigt für Scrolling).
- Weitere Treffer Button - Button zum Nachladen von weiteren Suchergebnissen.

## ii) Prozesse

Das Business Objekt hat sechs published Actions, die aus dem Prozess heraus aufgerufen werden können:

1. **Launch:** Startet den Chrome Browser und ruft die Webseite von Das Telefonbuch auf.
  - Input Parameter: keine
  - Output Parameter: keine
  - Voraussetzung: keine
  - Finaler Zustand: Chrome Browser ist gestartet und befindet sich auf der Startseite von [www.dastelefonbuch.de](http://www.dastelefonbuch.de).
2. **Search:** Führt eine Suche mit den übergebenen Parametern aus.
  - Input Parameter: Branche (text), PLZ (text)
  - Output Parameter: keine
  - Voraussetzungen: Die Anwendung ist gestartet und befindet sich auf der Startseite.
  - Finaler Zustand: Die Anwendung ist auf der "Results" Seite.

3. **Get Results:** Iteriert über die Ergebnisse der Seite und ermittelt jeweils Namen, Straße, Ort, PLZ und Branche.
- Input Parameter: Branche (Text).
  - Output Parameter: Results (Collection mit Einträgen bestehend aus Name (Text), City(Text), PLZ (Text), Street (Text), Branche (Text)).
  - Voraussetzungen: Die Anwendung ist auf der "Results"-Seite.
  - Finaler Zustand: Die Anwendung ist auf der "Results"-Seite.

Die Aktion **Get Results** iteriert über die Anzahl an Treffern die laut des Total Results Elements verfügbar sind. In Fällen in denen kein Ergebnis für die Input Parameter gefunden wird, gibt das Telefonbuch Ergebnisse für einen größeren Suchradius zurück. Diese Situation wird erkannt, wenn das **Keine Treffer Nachricht**-Element existiert und eine leere Collection wird zurückgegeben.

Da die Webseite nicht alle Einträge auf einmal, sondern beim Scrollen dynamisch lädt, wird nach jedem 10 Eintrag der interne *Show more* Prozess gestartet. Dieser navigiert zum **Footer** der Webseite und prüft, ob der **Weitere Treffer Button** existiert und drückt diesen gegebenenfalls.

Zum Auslesen der Daten wird der interne *Crawl Information* Prozess ausgelöst. Dieser erstellt eine neue Row für die Collection und liest die Entry-Elemente des Application-Models aus. Um die zum jeweiligen Eintrag gehörenden Felder zu identifizieren werden dynamische XPath-Abfragen verwendet, welche den Iterator Wert in die ID des jeweiligen HTML-Elements einfügen. Das folgende Beispiel liest den Namen eines Eintrags aus:

```
"//DIV[@id="&Chr(34)&"entry_"&[Iterator]&Chr(34)&"]//DIV[@class="&Chr(34)&"name  
&Chr(34)&"]/*//SPAN[1]"
```

In seltenen Fällen sind Einträge nicht vollständig und verfügen nicht über alle Felder. Daher wird vor jeder Anfrage die Existenz des Elements geprüft und diese entweder übersprungen oder ersetzt. So wird in den Fällen in denen kein **Ort** Element existiert der **Stadtteil** mittel der Methode **Extract Regex Match** ermittelt.

## 4. Close:

- Input Parameter: keine
- Output Parameter: keine
- Voraussetzungen: Chrome Browser ist geöffnet.
- Finaler Zustand: Chrome Browser ist geschlossen.

## 3.3.2 Verarbeitung

Ziel der Verarbeitung war die korrekte Erkennung der Unternehmen, die bereits einen konkreten Antrag auf Finanzhilfe gestellt haben, um ein erneutes Informationsschreiben zu vermeiden. Dazu wurde festgelegt, dass eine Queue mit den zu untersuchenden Unternehmen im vorhergehenden Sub-Prozess befüllt wird. Die Unternehmen, die benachrichtigt werden sollen, werden wiederum in eine zweite Queue eingefügt.

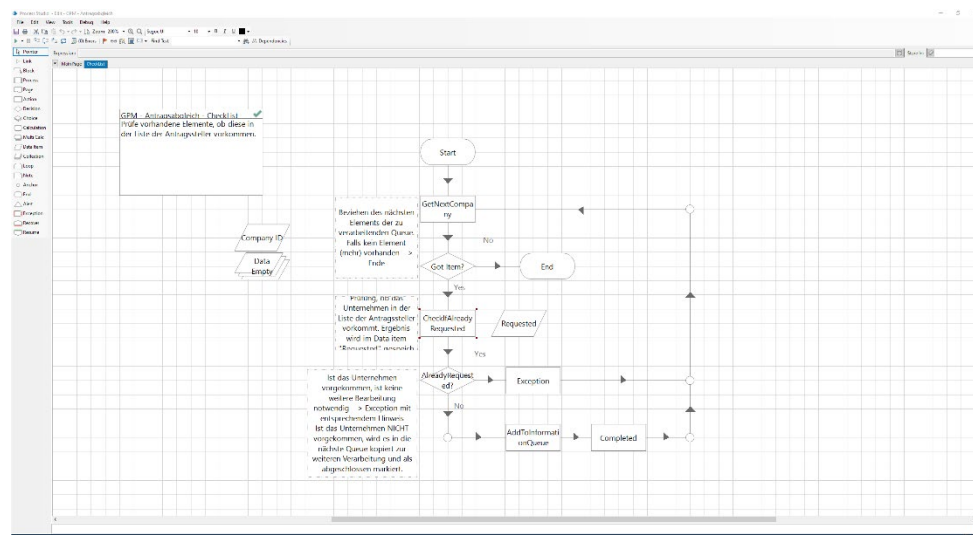


Abbildung 1 - Prozess "Abgleich mit Bestandskunden"

Da die Aufgabenstellung nicht die Suche nach den Unternehmen mit Antrag umfasste, wurde davon ausgegangen, dass eine entsprechende Collection bereits besteht und diese (ggf.) in den Prozess übergeben wird. Zunächst wird aus der Queue aller Unternehmen das nächste verfügbare Element herausgenommen. Sollte keines mehr vorhanden sein, endet der Prozess augenblicklich. Falls ein Element gefunden werden konnte, wird eine Überprüfung durchgeführt. Mit dem Utility "Collection Manipulation" (<https://github.com/blue-prism/collection-manipulation>) war eine Suche der aktuellen "Company ID" in der Collection "ComapnysWithRequest" in der Spalte "Company ID" durchgeführt.

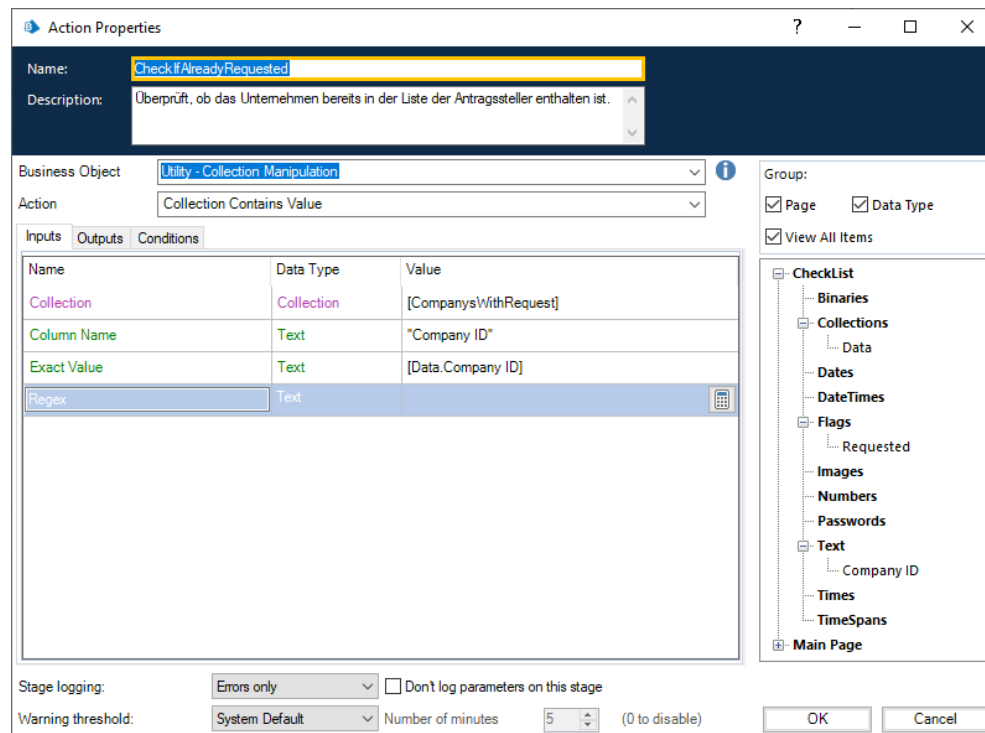


Abbildung 2 - Überprüfung der Collection

Ist das entsprechende Unternehmen gefunden worden, wird das Flag "Requested" auf `true` gesetzt. Anschließend wird dieses Flag überprüft. Sollte das Unternehmen gefunden worden sein, wird das Objekt in der Queue mit einer Exception abgeschlossen und das Nächste wird bearbeitet. Andernfalls wird das Element in die Queue "*CompaniesWithoutRequest*" kopiert und in der aktuellen Queue als "Completed" markiert.

Im Ergebnis werden nun die Elemente aus der Queue "*PotentialCompanies*" nacheinander in der Collection gesucht und bei Nichtexistenz in die nachfolgende Queue "*CompaniesWithoutRequest*" kopiert. Diese Lösung bietet den Vorteil, dass dieser Prozess gesondert auf mehreren Maschinen gestartet werden kann, um die vielen Unternehmen in der Queue parallel zu bearbeiten. Durch das Queue-Handling von BluePrism ist ausgeschlossen, dass bereits bearbeitete unternehmen erneut geprüft werden, da diese entweder als "Completed" oder als "Exception" markiert wurden. Aktuell bearbeitete Elemente sind blockiert und können ebenfalls nicht von einer zweiten Maschine bearbeitet werden. Dies erlaubt eine maximale Parallelisierung und eine extreme Zeitersparnis. Da die überprüften Unternehmen ebenfalls wieder in einer Queue zwischengespeichert werden, ist eine weitere Parallelisierung im Anschluss möglich.

### 3.3.3 Ausgabe

Die Collection aus der Queue „*CompanysWithoutRequest*“ wird für die Personalisierung der Anschreiben übernommen. Dadurch werden die Platzhalter im Anschreiben durch den Namen und die Adresse der Kunden ersetzt. Das Versanddatum wird auf „heute“ gesetzt. Anschließend wird jedes personalisierte Anschreiben als PDF in einen definierten Ordner hinterlegt, um ihm nächsten Subprozess verarbeitet zu werden.

Der Abschluss des Ausgabeteils besteht aus zwei Prozessen. Nach dem das Schreiben individualisiert wurde, müssen diese Daten mittels einer vorgegebenen Software (7z) komprimiert werden. Darüber hinaus leitet CoBee die Daten in einer E-Mail an den Auftraggeber weiter. Für den vorliegenden Prozess „Dateikomprimierung“ stellt CoBee eine Verbindung zum 7z-Programm über das Objekt „ATCS - Compression and Extraction“ her. Für die Anbindung des Objekts der Dateikomprimierung wurden die lokalen Input-Daten Prozesspfad, Eingangspfad und Zielpfad benötigt. Mittels diesen stellt das Objekt die Verbindung zu dem 7z-Programm her. Nach einem erfolgreichen Durchlauf des Subprozesses wird die komprimierte 7z-Datei, welche die individualisierten Schreiben enthält, unter dem Zielpfad gespeichert.

Wie bereits im ersten Teilprozess „Eingabe“ wird erneut eine Verbindung zum Google Mail Postfach über das POP-Protokoll aufgebaut. Dafür werden die gleichen Zugangsdaten, wie beim Empfang der E-Mails verwendet. Die 7z-Datei wird über einen lokalen Pfad iteriert und an die E-Mail angehängen. Wenn der Prozess erfolgreich durchgeführt wurde, findet der Auftraggeber eine E-Mail mit dem entsprechenden Anhang in seinem Postfach

Der Abschluss des Ausgabeteils besteht aus zwei Prozessen. Nach dem das Schreiben individualisiert wurde, müssen diese Daten mittels einer vorgegebenen Software (7z) komprimiert werden. Darüber hinaus leitet CoBee die Daten in einer E-Mail an den Auftraggeber weiter. Für den vorliegenden Prozess „Dateikomprimierung“ stellt CoBee eine Verbindung zum 7z-Programm über das Objekt „ATCS - Compression and Extraction“ her.



Für die Anbindung des Objekts der Dateikomprimierung wurden die lokalen Input-Daten Prozesspfad, Eingangspfad und Zielpfad benötigt. Mittels diesen stellt das Objekt die Verbindung zu dem 7z-Programm her. Nach einem erfolgreichen Durchlauf des Subprozesses wird die komprimierte 7z-Datei, welche die individualisierten Schreiben enthält, unter dem Zielpfad gespeichert.

Wie bereits im ersten Teilprozess „Eingabe“ wird erneut eine Verbindung zum Google Mail Postfach über das POP-Protokoll aufgebaut. Dafür werden die gleichen Zugangsdaten, wie beim Empfang der E-Mails verwendet. Die 7z-Datei wird über einen lokalen Pfad iteriert und an die E-Mail angehängen. Wenn der Prozess erfolgreich durchgeführt wurde, findet der Auftraggeber eine E-Mail mit dem entsprechenden Anhang in seinem Postfach.

### **3.4 Zusammenführung**

Die Erstellung des gesamten Prozesses innerhalb einer Arbeitsgruppe erforderte die Erstellung von Subprozessen und die anschließende zusammenführung in einen Gesamtprozess. Die Umsetzung gestaltete sich dabei prinzipiell sehr einfach, da zunächst ein neuer Prozess erstellt und dann die Subprozesse aufgerufen wurden. Lediglich die Ergänzung der Eingabecollection von Unternehmen mit Antrag musste ergänzt werden. Interessanter gestalteten sich hier die Schnittstellen zwischen den Subprozessen. Die Ausgabe des ersten Prozesses musste mit den Eingaben und Abhängigkeiten des zweiten Prozesses übereinstimmen. Gleiches galt auch zwischen dem zweiten und dem dritten Prozess.

### **3.5 Fragen und Probleme**

Während des Projektes sind sowohl bei der Vorbereitung als auch bei der Bearbeitung einige Probleme aufgetreten und Fragen aufgekomen, die im Folgenden gesammelt und erläutert werden. Sollte eine Lösung oder Beantwortung vorhanden sein, wird diese jeweils hier mit aufgeführt.

#### **3.5.1 Zusammenarbeit**

Blue Prism unterstützt keine gute Zusammenarbeit via einer Versionierungskontrolle wie z.B. Git, da die Dateien, die die Software im Rahmen der Entwicklung erstellt, im System gespeichert und nicht sinnvoll auszuwerten sind. Dadurch entstand die Frage nach der bestmöglichen Zusammenarbeit innerhalb eines Teams mit Blue Prism. Die Lösung hier ist

die unabhängige Entwicklung in eigenständigen Prozessen, die am Ende innerhalb eines Master-Prozesses aufgerufen werden.

### **3.5.2 Company ID**

Die Generierung der eindeutigen ID erzeugte die Frage nach dem Zeitpunkt der Erstellung. Der Vorteil der Verwendung der ID war vor allem beim Abgleich zu den Unternehmen, die bereits einen Antrag auf Corona-Hilfe gestellt haben, zu erkennen. Statt des Abgleichs zwischen Name, Adresse, PLZ, Ort, musste lediglich eine ID verglichen werden. So fiel die Entscheidung zur Generierung der ID auf die Suche nach dem Unternehmen im Telefonbuch.

### **3.5.3 Definierung von Schnittstellen**

Die unabhängige Arbeit der Teammitglieder in den Teilbereichen erforderte die Übergabe bzw. Bereitstellung von Daten. Bei der Zusammenführung der Teilprozesse wurde bemerkt, dass einige Schnittstellen unzureichend definiert waren, sodass zunächst eine Kompatibilität hergestellt werden musste, um die Prozesse harmonisch in Verbindung zu bringen.

## 4 Auswirkungen auf den Referenzprozess

### 4.1 Prozessveränderung

Die im Bankmodell enthaltenen Referenzprozesse „Finanzieren“ und „Vertrieb“ stellen zwar weiterhin die Grundlage für den Case „Recherche zur Ermittlung von Berechtigten für Finanzhilfen“ dar, sie müssen allerdings aufgrund Anzahl der potenziellen Berechtigten und der damit notwendigen Automatisierung der einzelnen Prozessschritte in ihren wesentlichen Teilen angepasst werden. Die Corona-Hilfen werden von den Betroffenen in der Regel dringend benötigt, was die Automatisierung auch aus der Kundenperspektive notwendig macht. Durch die Bereitstellung der Finanzierungshilfen durch die KfW sowie die umfangreiche Risikoübernahme durch die KfW wird der Referenzprozess „Finanzieren“ zusätzlich verändert.

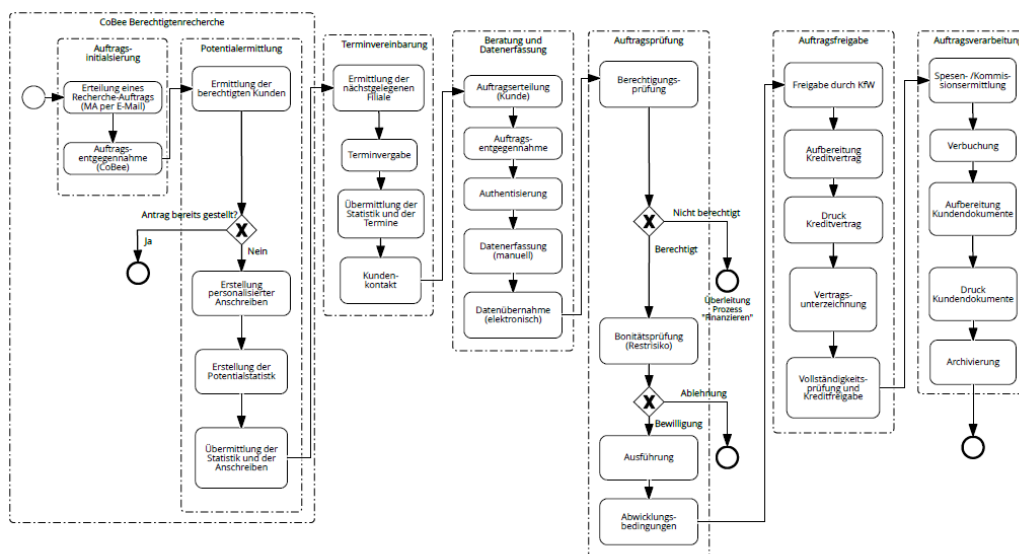


Abbildung 3 - Referenzprozess

Der angepasste Prozess ist in Abbildung 1 „Angepasster Referenzprozess für Corona-Hilfen“ dargestellt. Bereits die Auftragserteilung weicht vom klassischen Finanzierungsprozess ab, denn hier wird der Auftrag nicht durch den Kunden selbst, sondern der zuständige Mitarbeiter der Fachabteilung für Corona-Hilfe innerhalb der Commerzbank beauftragt die als CoBee bezeichnete Robotics-Unterstützung das Potential für Corona-Hilfen im Kundenbestand zu ermitteln. Diese führt den Auftrag durch die Abfrage des bestandsführenden Systems nach Kunden bestimmter Branchen in den definierten Postleitzahlbereichen sowie den Abgleich mit der Liste der bereits gestellten Anträge auf Corona-Hilfe durch. Sofern ein Kunde gefunden wird, der die Voraussetzungen für einen

entsprechenden Antrag erfüllen könnte, wird ein personalisiertes Anschreiben erstellt und ein Bericht erzeugt. Beides wird an den zuständigen Mitarbeiter, der den Auftrag ausgelöst hat, übermittelt. Die Automatisierung dieser Prozessschritte ist Fokus der Gruppe I. Um den Prozess jedoch vollends zu modellieren und in den Kontext des Bankmodells integrieren zu können, müssen auch nachfolgende Prozessschritte berücksichtigt werden.

Als nächstes erfolgt die ebenfalls automatisierte Vergabe von Terminen an die zuvor ermittelten Kunden mittels einer weiteren CoBee. Den Kundenkontakt sowie die anschließende Beratung kann aktuell noch nicht automatisiert erfolgen, weswegen ab hier auf die Expertise der Kundenberater in den einzelnen Filialen der Commerzbank zurückgegriffen wird. Darüber hinaus wird vom Kundenberater auch die Berechtigung des Kunden für die Corona-Hilfen geprüft. Sofern die Berechtigung vorliegt, muss je nach geeignetem Unterstützungsprogramm die Bonität des Kunden für das möglicherweise vorliegende Kreditrisiko geprüft werden. Dieser Schritt entfällt, falls die Bereitstellung der Corona-Hilfen als KfW-Schnellkredit 2020 erfolgen kann, denn hier übernimmt die KfW das Kreditausfallrisiko zu 100%. Die Vergabe der Unterstützung muss allerdings durch die KfW in einem weiteren Schritt bestätigt werden, was ebenfalls eine Abweichung vom klassischen Referenzprozess „Finanzieren“ darstellt. Die restlichen Schritte bis zur Beendigung des Prozesses entsprechen hingegen dem ursprünglichen Referenzprozess.

Wie der Abbildung 1 zu entnehmen ist, wurde für die Remodellierung des Referenzprozesses auf die Methode „Aggregation“ zurückgegriffen, indem der Referenzprozess „Finanzieren“ mit dem Referenzprozess „Vertrieb“ verschmolzen wurde. Dies ist in den durch die Automatisierung veränderten Reihenfolge der einzelnen Prozessschritte ursächlich. Aufgrund der besonderen Art der Finanzierung mittels staatlicher geförderter Programme wurden die einzelnen Elemente in den Prozessschritten „Auftragsprüfung“ und „Auftragsfreigabe“ konkretisiert und verändert. Diese Art der Remodellierung wird in der Literatur als Instanziierung bzw. Spezialisierung bezeichnet.

## **4.2 Wertbeitrag**

Durch die Automatisierung der einzelnen Prozessschritte können Personalkosten eingespart werden. Um den monetären Wert der Investition in das entwickelte IT-System zu bewerten, kann entweder das Verfahren „Time Saving Time Salary“, welches den Wert der eingesparten Arbeitszeit der Mitarbeiter betrachtet, oder das hedonistische Verfahren,

welches auf der Verlagerung der freigewordenen Mitarbeiterkapazitäten in Richtung anspruchsvollerer Tätigkeiten beruht, eingesetzt werden. Letzteres erfordert allerdings tiefere Kenntnis der Aufgaben und Potentiale der eingesparten Arbeitskräfte, die uns fehlt, weswegen wir auf das einfachere Verfahren zurückgreifen müssen (vgl. Sassone 1987).

Das „Time Saving Time Salary“ Verfahren basiert auf der Prämisse, dass die Personalkosten eines Mitarbeiters dem Wert seiner Arbeit entsprechen. Da die automatisierten Tätigkeiten kein Vorwissen erfordern und kaum Fehlerpotential bergen, könnten sie auch von Zeitarbeitskräften ohne eine fachliche Ausbildung erledigt werden. Diese würde man nur gesondert für die dedizierten Aufgaben einstellen, sodass die Prämisse eingehalten wird. Für die Vereinfachung der Berechnung wird unterstellt, dass die Arbeitskräfte den in 2020 gültigen Mindestlohn von 9,35 EUR pro Stunde vergütet bekommen. Die Lohnnebenkosten werden auf 20% geschätzt. Folglich ist eine eingesparte Stunde 11,22 EUR wert.

Die Ermittlung der eingesparten Arbeitszeit erfolgt nur auf der Grundlage der von Gruppe I automatisierten Aufgaben. Es handelt sich im Wesentlichen um drei Schritte:

1. Recherche im Bestandskundensystem (Telefonbuch)
2. Abgleich mit der Liste der bereits gestellten Anträge
3. Erstellung der personalisierten Anschreiben

Jeder der Schritte wurde je 15 Mal von zwei Testpersonen ausgeführt. Die benötigte Zeit wurde mit einer Stoppuhr gemessen und als endgültige Wert der Median der gemessenen Zeiten herangezogen. So beträgt die eingesparte Arbeitszeit für die Recherche 28 Sekunden, für den Abgleich mit der Liste – 52 Sekunden und für die Erstellung des Anschreibens – 13 Sekunden. Insgesamt konnten durch die Automatisierung somit 93 Sekunden pro potenziellen Antragsteller eingespart werden. Es muss jedoch angemerkt werden, dass die Ausführungsgeschwindigkeit am Ende der Stichprobe deutlich niedriger ausfiel als am Anfang. Dieser Effekt könnte allerdings mit einer zunehmenden Anzahl an Wiederholungen aufgrund der sehr niedrigen Abwechslungslosigkeit der Aufgaben wieder umkehren.

Um die gesamte Zeit, die die Commerzbank für die Ermittlung der potentiellen Berechtigten für die Corona-Hilfen müsste die eingesparte Zeit mit der Anzahl der betroffenen Kunden der Commerzbank multipliziert werden. Leider veröffentlicht die Commerzbank keine Zahlen zu der Anzahl betreuten Firmenkunden. Laut IfM Bonn gab es im Jahr 2018 3,47

Millionen kleine und mittlere Unternehmen (KMU). Die Commerzbank ist zwar die zweitgrößte Bank in Deutschland, jedoch ist der Bankenmarkt in Deutschland stark fragmentiert und die Commerzbank als Großbank nicht auf KMU spezialisiert, weswegen wir höchstens von einem Marktanteil von 2% ausgehen. Somit hätte die Commerzbank 69.400 potenzielle Berechtigte. Wir nehmen ferner an, dass rd. 30% dieser Kunden bereits proaktiv einen Antrag gestellt haben. Somit müssten die automatisierten Tätigkeiten 48.580 Mal wiederholt werden. Folglich werden rd. 1.255 Arbeitsstunden eingespart. Somit beträgt der monetäre Wert der Investition in die erstellte CoBee 14.081,10 EUR.

### **4.3 Strategische Bedeutung**

Über den aktuellen UseCase hinaus können die einzelnen Bausteine der erstellten Anwendung aufgrund des abstrakten Charakters wiederverwendet werden. Unter anderem kann im Kundenbestandsystem auch nach anderen Merkmalen gesucht werden, um ebenfalls die Potentiale zu ermitteln. Abgleich mit den bereits gestellten Anträgen ist in jedem Vorgang sinnvoll, um dem Kunden eine unnötige Ansprache zu ersparen.

Durch den Einsatz der CoBee schafft die Commerzbank die Grundlage für weitere Prozessautomatisierung, die auch dem geplanten Stellenabbau zu Gute kommen wird. Dabei werden von den CoBees (noch) keine vollumfänglichen Entscheidungen getroffen. Diese werden allerdings dem Kundenberater und dem Kunden durch aktive Übermittlung von Entscheidungsinformationen erleichtert.

