

**Visualisierung von Aufgabenrelation
innerhalb der Process Engine
Automic[®] Automation
Anhand archivierter Daten**

Bachelor-Thesis
zur Erlangung des akademischen Grades B.Sc.

Jan Thomas Gundlach
2038884



Hochschule für Angewandte Wissenschaften Hamburg
Fakultät Design, Medien und Information
Department Medientechnik

Erstprüfer: Prof. Dr. Larissa Putzar

Zweitprüfer: Dipl.Ing. (fh) Thorsten Wagener

Hamburg, 31. 8. 2021

Inhaltsverzeichnis

1	Einleitung	6
1.1	Motivation	8
1.2	Problemstellung	9
1.3	Zielsetzung	9
1.3.1	Anforderung	10
1.3.2	Abgrenzung	10
1.4	Struktur der Arbeit	11
2	Grundlagen	12
2.1	Was ist Business Process Management?	12
2.2	Was ist eine Process Engine?	13
2.3	Automic® Automation	14
2.3.1	Workflows	16
2.3.2	Jobs	16
2.3.3	Ausführungen	17
2.3.4	Reports	18
2.3.5	Reorganisation	18
2.4	Automic® Archive Browser	19
2.5	Rechtliche Grundlagen	19
2.5.1	VAIT	20
2.5.2	ITSiG / BSI-KritisV	26
2.5.3	DSGVO	27
2.6	Java Implementierung	28
2.7	GUI-Framework	29

3	Analyse	30
3.1	Automic® Archive Browser	30
3.1.1	Ermittlung von Benutzeranforderungen	30
3.1.2	Weiterentwicklung	36
3.1.3	Datenstruktur	39
3.2	Auswahl der Java Implementierung	43
3.2.1	Amazon Corretto	45
3.2.2	Azul Zulu	46
3.2.3	Eclipse Temurin	46
3.2.4	IBM OpenJDK	47
3.2.5	Oracle Java SE	48
3.2.6	Red Hat OpenJDK	48
3.2.7	Ergebnis	49
3.3	Auswahl des Java GUI-Framework	50
3.3.1	Swing	51
3.3.2	JavaFX	51
3.3.3	SWT	52
3.3.4	Ergebnis	53
3.4	Java Dateieingangsverarbeitung	54
3.4.1	Klasse/Methode für Protokolle	54
3.4.2	Klasse/Methode für Ausführungsdaten	55
4	Implementierung	58
4.1	Paketierung	58
4.2	Klasse ArchiveData	58
4.2.1	Methoden	60
4.3	Klasse ArchiveTreeTable ¹	61
4.3.1	start	62
4.4	Datenstruktur	64
4.4.1	Aufbau der Baumstruktur	64
4.4.2	Darstellung der Baumstruktur	65
4.5	Klassendiagramm	65
4.6	Ergebnis	65

5	Fazit und Ausblick	68
5.1	Fazit	68
5.2	Ausblick	68
5.2.1	Bugs	68
5.2.2	Features	69
5.2.3	DORA	69
	Abbildungsverzeichnis	70
	Tabellenverzeichnis	75
	Listings	76
	Literaturverzeichnis	78

Abstract

This thesis deals with the requirement and development of a relation-oriented retrieval tool for archived log data from a process engine. The implementation is based on the Automic® Automation Engine. For the requirement the legal background is worked out using the insurance industry as an example and the structure of the process engine is described. The basis for the development and the scope of functions for the search tool is a fundamental part of the thesis as well as the implementation. The result is the prototype of a utility, with which the dependencies of archived protocols of a process engine can be traced. The resulting concept is based on a combination of Eclipse Temurin and OpenJFX. Finally, an outlook on future requirements and the further development of the prototype is described.

Zusammenfassung

Diese Arbeit behandelt die Anforderung und Entwicklung eines relationsorientierten Recherchewerkzeugs für archivierte Protokolldaten aus einer Process Engine. Die Umsetzung erfolgt am Beispiel der Automic® Automation Engine. Für die Anforderung wird der rechtliche Hintergrund am Beispiel des Versicherungswesens erarbeitet und der Aufbau der Process Engine beschrieben. Thematisiert wird die Auswahl der Grundlagen für die Entwicklung und des Funktionsumfangs für das Recherchewerkzeug mit der daraus folgenden Implementierung.

Ergebnis ist der Prototyp eines Dienstprogramms, mit dem die Abhängigkeiten von archivierten Protokollen einer Process Engine dargestellt und mit dem über diese traversiert werden können. Dabei setzt das entstandene Konzept auf eine Kombination aus Eclipse Temurin und OpenJFX. Abschließend wird ein Ausblick auf kommende Anforderungen und die Weiterentwicklung des Prototyps gegeben.

1 Einleitung

Wenn Mitarbeiter mittelständischer und größerer Unternehmen oder Organisationen morgens mit der Arbeit beginnen, werden viele Dinge für selbstverständlich genommen. Zügiges Öffnen und Anmelden an zentralen Systemen. Arbeitsaufträge, die kategorisiert oder chronologisiert bereitstehen. Der Aufruf von Unterlagen, die zur Bearbeitung entsprechender Aufträge benötigt werden. Auswertungen, die Vortags in Auftrag gegeben wurden. Zugriff auf historische Daten von erledigten Aufträgen und vieles mehr.

Im Hintergrund müssen dafür an diversen Stellen jedoch zum Teil ungeahnt komplexe und teils miteinander verbundene Abläufe erfolgreich stattgefunden haben. Beispielsweise müssen zur Definition von Arbeitsaufträgen Eingangsdaten wie Posteingänge aus verschiedenen Medienkanälen wie Briefpost, E-Mail oder anderen Kontaktschnittstellen verschiedenster Applikationen zusammengeführt und vorverarbeitet werden. An der Vorverarbeitung sind häufig eine Vielzahl von Applikationen beteiligt, die die Daten durchlaufen müssen. All diese Prozessschritte müssen miteinander verknüpft werden, damit die richtigen Arbeitsaufträge später beim zuständigen Sachbearbeiter landen oder bereits vollautomatisiert bearbeitet werden können.

Fehler in diesen Prozessketten müssen, wenn möglich, automatisch behandelt werden. Falls nicht, muss die Verarbeitung automatisch angehalten werden, um eine manuelle Behebung zu ermöglichen. Die Verantwortlichen müssen über den Fehler und dass ein manuelles Eingreifen notwendig ist, umgehend in Kenntnis gesetzt werden. Nach erfolgreicher Behebung muss die Verarbeitung genau an dem Punkt wieder aufgenommen werden können, an dem sie stehen geblieben ist. Falls jedoch der Fehler nicht vollständig behoben wurde oder weitere Probleme auftreten, muss der Schritt beliebig oft wiederholt werden können.

Um zentrale Systeme zügig aufzurufen und auch historische Daten unverzüglich zur Verfügung zu stellen, müssen die Datenbestände dieser Systeme regelmäßig reorgani-

1 Einleitung

siert und nötigenfalls archiviert werden können.

All dies erfordert eine übergeordnete Einheit, in der die Prozessschritte geplant, ausgeführt, mit einander verknüpft und überwacht werden. Damit ist es aber noch nicht getan. Hintergrund ist, dass Unternehmen oder Organisationen in ihrer Verarbeitung häufig regulatorischen Auflagen oder zumindest gesetzlichen Pflichten unterliegen. Das bedeutet, dass es Vorgaben gibt, wie eine Verarbeitung zu erfolgen hat oder was bei einer Verarbeitung zu beachten ist. Diese Vorgaben können je nach Sektor und Branche unterschiedlich sein. Insbesondere für Unternehmen oder Organisationen, die durch ihre Tätigkeit zur kritischen Infrastruktur ¹ gehören, können die Vorgaben restriktiv sein. Die Unternehmen oder Organisationen sind teilweise dazu verpflichtet, die Einhaltung dieser Auflagen und Pflichten regelmäßig durch zertifizierte Dritte überprüfen zu lassen. Diese Prüfungen können seitens der zuständigen Anstalt oder Behörde auch unregelmäßig anlasslos oder anlassbezogen erfolgen. Entsprechend muss die Verarbeitung jederzeit nachweisbar und überprüfbar sein. Parallel ist es auch im eigenen Interesse eines Unternehmens oder einer Organisation, eine nachweisbare und überprüfbare Verarbeitung vorweisen zu können. In juristischen Auseinandersetzungen mit Kunden oder Partnern müssen bestimmte Vorgänge ggf. gerichtsfest nachgewiesen werden können, um unbegründete Forderungen oder Regressansprüche abwehren zu können.

Um einen Vorgang nachweisen zu können, muss es zum Einen eine dokumentierte Richtlinie geben, wie bzgl. des Vorganges verfahren wird. Zum Anderen sollte der Geschäftsprozess anhand der technischen Artefakte des Vorganges modelliert werden können. Bei einer Überprüfung muss nachgewiesen werden, dass die Richtlinie durch die Geschäftsprozessmodellierung (vgl. 2.1) in dem konkreten Fall auch so wie beschrieben umgesetzt wurde. Hierfür muss die Durchführung des Geschäftsprozesses protokolliert werden und die Protokolle müssen bei Reorganisation unverändert und unveränderbar abgelegt werden.

Im Prüfungsfall kann so theoretisch eine bestimmte Durchführung eines Geschäftsprozesses innerhalb des Aufbewahrungszeitraumes anhand des Protokolls nachgewiesen werden. In der Regel besteht ein Geschäftsprozess aber je nach Detaillierungsgrad aus mehreren Teilprozessen, die ebenfalls aus Teilprozessen bestehen können und am Ende

¹Quelle: Verordnung zur Bestimmung Kritischer Infrastrukturen nach dem BSI-Gesetz (BSI-Kritisverordnung - BSI-KritisV), <https://www.gesetze-im-internet.de/bsi-kritisv/BJNR095800016.html>, Aufruf: 12.07.2021

in einzelne Aktivitäten unterteilt sind. Jeder Geschäftsprozess, jeder Teilprozess und jede Aktivität verfügt über ein eigenes Protokoll mit entsprechendem Detaillierungsgrad. In einem komplexen Geschäftsprozess ist die Einhaltung der Vorgaben anhand des Protokolls des Geschäftsprozesses selbst aber ggf. nicht vollständig nachvollziehbar. Für eine Prüfung werden daher nicht selten auch die Protokolle der Teilprozesse und Aktivitäten benötigt. Eine Prüfung erfolgt in der Regel erst, wenn die Protokolle reorganisiert und archiviert wurden. Je nach Reorganisationssystematik der Process Engine (vgl. 2.2) können die Relationen an sich zwischen Geschäftsprozess, Teilprozess und Aktivität bei der Reorganisation und Archivierung verloren gehen. Bietet das Betrachtungswerkzeug für die archivierten Protokolle keine Möglichkeit, die Relationen neu zu ermitteln und über diese zu traversieren, müssen die Relationen im Prüfungsfall im Zweifel manuell recherchiert werden.

1.1 Motivation

Als Mitverantwortlicher für die zentrale Process Engine (vgl. 2.2) Automic® Automation ist der Autor Ansprechpartner zu Fragen zum tatsächlichen Ablauf bestimmter Durchführungen von Geschäftsprozessen und Ermittlung von Ablaufnachweisen. Sofern die Protokolle zu den angefragten Abläufen bereits reorganisiert und archiviert wurden, entsteht durch die Recherche von Protokollen zu Teilprozessen und Aktivitäten eines angefragten Geschäftsprozesses zurzeit gegebenenfalls ein hohes Arbeitsaufkommen. Im Prüfungsfall kann die Recherche auf Verlangen des Prüfers auch während einer Prüfung und im Beisein des Prüfers durchzuführen sein. Ein hohes Arbeitsaufkommen wirft in der Regel kein gutes Licht auf die Nachweisbarkeit und Nachvollziehbarkeit bzgl. der Durchführungen von Geschäftsprozessen innerhalb der Prüfung.

Innerhalb der Process Engine (vgl. 2.2) Automic® Automation existiert für nicht reorganisierte Protokolle die Möglichkeit zu einer Ausführung eines Geschäftsprozesses (vgl. 2.3.1) über die Relationen zu Teilprozessen (vgl. 2.3.2) bis hin zu einer Aktivität (vgl. 2.3.3) zu traversieren. Dies ermöglicht eine deutlich übersichtlichere und schnellere Recherche durch die Protokolle komplexer Geschäftsprozesse. Daraus entstand der Anreiz, die Recherchemöglichkeiten der Process Engine (vgl. 2.2) Automic® Automation und des Betrachtungswerkzeuges für archivierte Protokolle zu

harmonisieren.

1.2 Problemstellung

Der Automic® Archive Browser, das bestehende Betrachtungswerkzeug für archivierte Protokolle der Process Engine (vgl. 2.2) Automic® Automation, ist für eine flache Recherche ausgelegt. Die den archivierten Protokollen zu Grunde liegenden Datenstrukturen sind entsprechend aufgebaut. Der Automic® Archive Browser bietet keine zugängliche oder anwendbare API, die zum Einlesen der Protokolle wiederverwendet oder auf die aufgebaut werden könnte.

Parallel lässt sich die Abbildung der Relationen, die sich durch Ausführung verschiedener Geschäftsprozesse zwischen Teilprozessen und Aktivitäten in den Automic® Automation Protokollen ergibt, bspw. nicht in einem binären Suchbaum überführen. Hintergrund ist, dass sich über alle Relationen hinweg keine allgemeingültige oder abstrakte Entscheidungsregel ableiten lässt, in welchen Knoten ein Protokoll eingefügt werden muss. Außerdem sind pro Teilbaum theoretisch beliebig viele Knoten möglich.

Daneben ergeben sich durch die Entwicklung in Java mehrere GUI-Frameworks, mit denen die grafische Darstellung realisiert werden kann. Diese verfügen über ganz unterschiedliche Vor- und Nachteile, die gegeneinander abgewägt werden müssen. Dadurch, dass die Protokollstände mehrerer Tage schnell eine fünf- oder sechsstellige Protokollanzahl erreichen können, muss das Framework grundsätzlich zur stabilen und performanten Darstellung großer Datenmengen geeignet sein.

Unter den beschriebenen Aspekten wird die praktische Umsetzung eines relationsorientierten Betrachtungswerkzeuges für archivierte Protokolle anhand von Automic® Automation vorgestellt. In diesem Zusammenhang wird die grundlegende Entwicklung des Betrachtungswerkzeuges durchgeführt.

1.3 Zielsetzung

Ziel der Arbeit ist eine relationsorientierte Darstellung von archivierten Ausführungsprotokollen und die Implementierung der dafür notwendigen Grundfunktionalitäten. Hierzu gehört die Auswahl einer passenden Umgebung als auch Frameworks und

das Erarbeiten einer entsprechenden Architektur. Außerdem soll der rechtliche Hintergrund am Beispiel des Versicherungswesens betrachtet werden, aus dem sich die Anforderung für ein relationsbasiertes Recherchewerkzeug für historische Protokoll-daten ableitet. Bei der Umsetzung wird besonderer Wert auf die Benutzerfreundlichkeit gelegt unter der Berücksichtigung des angewendeten Frameworks. Zudem soll auf Basis einer zukunftsfähigen Programmiersprache ein geringer Wartungsaufwand der Applikation erreicht werden.

1.3.1 Anforderung

Es soll die bestehende Grundlage des Automic[®] Archive Browser betrachtet als auch der geforderte Funktionsumfang ermittelt werden. Parallel sollen rechtliche Hintergründe anhand des Versicherungswesens aufbereitet werden. Davon ausgehend ist der technische Bedarf für ein relationsorientiertes Recherchewerkzeug für archivierte Ausführungsprotokolle zu definieren. Darauf aufbauend steht die Implementierung der Grundfunktionalitäten für die Recherche bestimmter Protokolle komplexer Geschäftsprozesse und das Traversieren über deren Abhängigkeiten im Mittelpunkt. Hierzu zählen das Einlesen von archivierten Daten, die Darstellung der Relationen der archivierten Daten sowie die Möglichkeit, nach bestimmten Informationen zu suchen.

1.3.2 Abgrenzung

Diese Arbeit ist explorativ. Der Fokus liegt primär auf der Struktur der archivierten Daten der Process Engine (vgl. 2.2) Automic[®] Automation. Die rechtlichen Aspekte werden mit Blick auf das Versicherungswesen betrachtet.

Es können innerhalb des zeitlich begrenzten Rahmens der Bachelorarbeit keine umfangreichen Benutzer- oder Systemtests durchgeführt werden. Grafisch finden die Benutzerschnittstellen wie auch sämtliche Fensterkomponenten im Standardstil des ermittelten GUI-Frameworks Anwendung.

Vorrangig geht es darum, die Grundlage für ein performantes, stabiles, erweiterbares, portierbares und einfach wartbares Recherchewerkzeug zu erstellen, um die Grundfunktionen zum Ermitteln bestimmter Ausführungsprotokolle und die Nachverfolgung deren Abhängigkeiten zu realisieren.

1.4 Struktur der Arbeit

2 Grundlagen

2.1 Was ist Business Process Management?

Zu Business Process Management (BPM) gibt es eine Vielzahl an unterschiedlichen Definitionen. Unter anderen gibt es eine Definition von der European Association of Business Process Management (EABPM), die sich aus verschiedenen europäischen Vereinen zusammensetzt, um ein international einheitliches Verständnis von BPM zu schaffen. In der letzten Version und Ausgabe der deutschen Fassung des Leitfadens „BPM Common Body of Knowledge (CBOK®)“ (vgl. [EABPM 2014](#)) der EABPM wird BPM wie folgt definiert:

„Die englische Bezeichnung „Business Process Management“ oder BPM wird synonym verwendet für Geschäftsprozessmanagement oder auch einfach Prozessmanagement. Als Prozess wurde oben eine Reihe von festgelegten Aktivitäten definiert, die von Menschen oder Maschinen ausgeführt werden, um ein oder mehrere Ziele zu erreichen. Letztlich geht es darum, einen Kundennutzen zu schaffen und damit auch für das Unternehmen Wert zu generieren.

Business Process Management (BPM) ist ein systematischer Ansatz, um sowohl automatisierte als auch nicht-automatisierte Prozesse zu erfassen, zu gestalten, auszuführen, zu dokumentieren, zu messen, zu überwachen und zu steuern und damit nachhaltig die mit der Unternehmensstrategie abgestimmten Ziele zu erreichen.

[...]

BPM umfasst Strategien, Ziele, Kultur, Organisationsstrukturen, Rollen, Grundsätze, Regeln, Methoden und IT-Werkzeuge, um a) End-to-End-Prozesse zu analysieren, zu entwerfen, einzuführen, zu steuern und kon-

tinuierlich zu verbessern und b) eine Prozess-Governance aufzubauen.“
(EABPM 2014: 62)

Im BPM wie auch im einfachen Prozessmanagement geht es entsprechend darum, nicht nur Prozessfragmente zu bewerten, sondern Prozesse ganzheitlich zu verstehen und zu verbessern. Dies geschieht in einer kombinierten Anwendung sowohl aus der organisatorischen Sicht als auch unter informationstechnischen Gesichtspunkten.

2.2 Was ist eine Process Engine?

Eine Process Engine, auch Workflow Engine genannt (vgl. [Freund, Rücker 2019](#)), dient zur Prozessautomatisierung definierter Prozesse aus dem BPM. Die Process Engine ist dabei das zentrale System, welches technische Prozessmodelle ausführt. Hervorzuheben ist hierbei, dass die Prozessautomatisierung nicht in jedem Fall eine Automatisierung des gesamten Prozesses bedeutet. Die Process Engine steuert den Prozess vielmehr, indem sie a.) zu erledigende Aufgaben bspw. an Sachbearbeiter delegiert und das Resultat weiterverarbeitet (Human Workflow Management) und b.) Schnittstellen interner/-externer Informationsdienste aufruft (Dienst-Orchestrierung). So bleibt ein automatisierter Prozess durch die Prozessbeteiligten beeinflussbar. Anhand des technischen Prozessmodells als auch des Resultates eines Service-Aufrufes oder einer beendeten Aufgabe entscheidet die Process Engine, unter welchen Gegebenheiten welche Folgeaufgabe oder -aufruf ausgeführt wird.

Abb. 2.1 veranschaulicht die Vorgehensweise verallgemeinert.

Abstrakt kann man Prozessautomatisierung als eine Form von Softwareentwicklung bezeichnen, in der die Process Engine der Interpreter und der Quellcode das technische Prozessbild sind. Dabei ist zu beachten, dass eine Process Engine für die Abbildung von Prozesslogik entwickelt ist. Sie verfügt über viele typische, fertige Komponenten und Eigenschaften, die für die technische Umsetzung eines Prozessmodells benötigt werden, welche ansonsten mittels höherer Programmiersprache (High-level language) (HLL) aufwendig entwickelt werden müssten. Dennoch kann mit einer Process Engine keine allgemeine Softwareentwicklung erfolgen, da ihre Sprachmittel und Komponenten auf die Abbildung von Prozessmodellen begrenzt sind.

Eine Process Engine bietet die Möglichkeit, jede Form von End-to-End-Prozessen

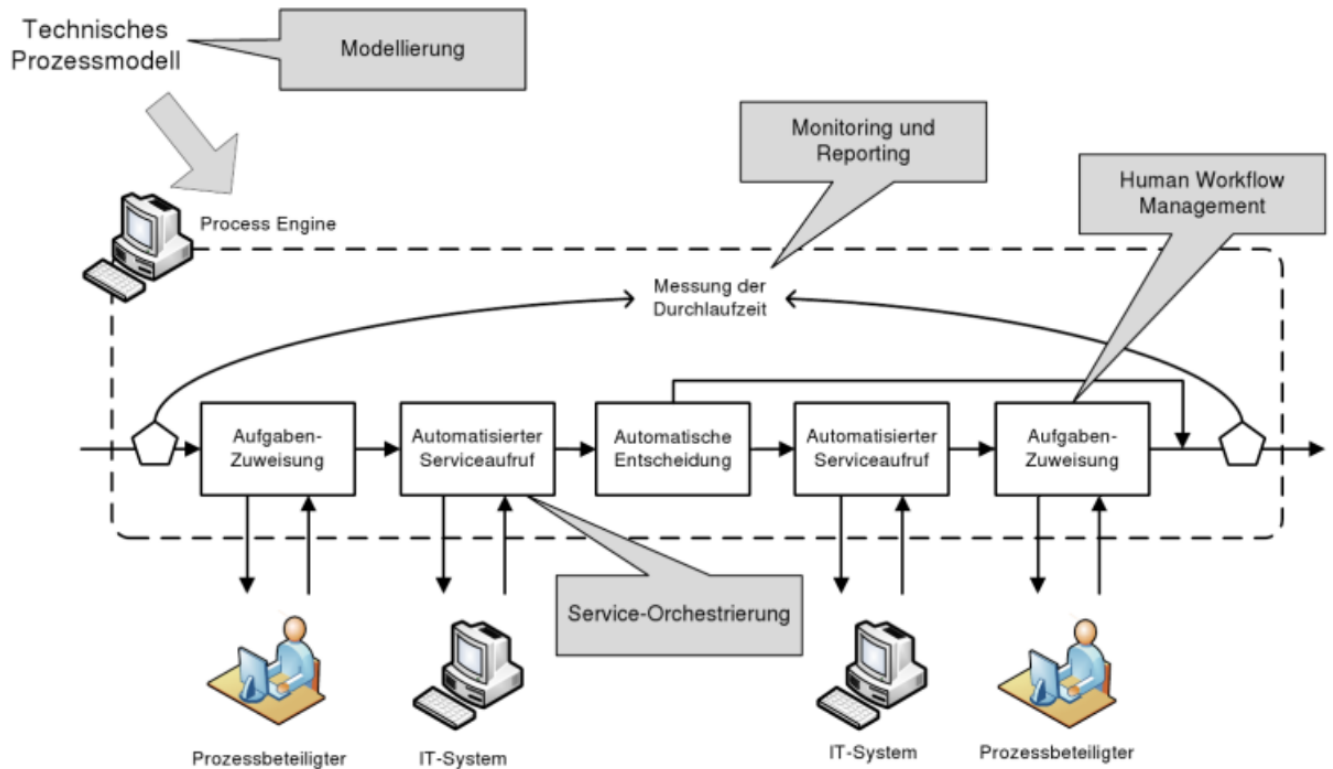


Abbildung 2.1: Prozessautomatisierung mit einer Process Engine¹

plattform- und standortunabhängig umzusetzen. Damit vereint sie Workflow Management und Anwendungsintegration. Durch die Steuerung der Geschäftsprozesse bietet die Process Engine zugleich einen Überblick in Echtzeit über selbige. Sie verfügt über Informationen zur Ausführungsdauer eines gesamten Prozesses oder eines Teilprozesses sowie den Status jeder Aufgabe. Sie liefert damit Metriken für das Controlling der unternehmensweiten Prozesslandschaft. Auf dieser Basis kann die Process Engine zugleich eine stets aktuelle Prozessdokumentation für Management als auch IT bieten.

2.3 Automic® Automation

Automic® Automation (vgl. [Broadcom 2021](#)) ist eine Gruppe zusammengehöriger Softwareprodukte, mittels der plattformübergreifend IT-bezogene Aufgaben automatisiert und dokumentiert werden können. Die zugehörigen Softwareprodukte sind im

¹Quelle: ([Freund, Rücker 2014](#): 7, Abb. 1.2)

Allgemeinen:

- Die Automation Engine als Process Engine bestehend aus (vgl. [Broadcom 2021](#)):
 - Einem primären Arbeitsprozess (PWP), der zentrale, nicht verteilbare Aufgaben wie Verwaltungsaufgaben durchführt.
 - Einem oder mehrere Arbeitsprozesse (WP), die alle sonstigen, eingeplanten Aufgaben ausführen oder Serverrollen übernehmen. Jeder WP kann bei Bedarf zum PWP werden.
 - Einem oder mehrere Kommunikationsprozesse (CP), die für die Kommunikation zwischen den Prozessen, Agenten auf Drittsystemen/-diensten und Schnittstellen zuständig sind.
 - Optional einem oder mehreren Dialogprozessen (DWP), die für die Meldungen im Automic[®] Web Interface (AWI) verwendet werden.
 - Einem oder mehreren Java-Arbeitsprozessen (JWP), die für speziell in Java entwickelte Dienste zur Verfügung stehen.
 - Einem Java-Kommunikationsprozess (JCP) für spezielle Dienste- und Schnittstellen wie die REST-API.
- Agenten zur Steuerung dritter Systeme oder Dienste.
- Das AWI als Benutzerschnittstelle.
- Der Plugin Manager zum Verwalten von Packs im AWI.
- Der Package Manager zum Verwalten von Packs auf der Kommandozeile.
- Der Action Builder zum Erstellen von Packs.
- Analytics On Premise für individuelle Statistiken.
- Verschiedene Dienstprogramme zur Verwaltung von Automic[®] Automation. Relevant innerhalb dieser Arbeit sind dabei:
 - AE DB Archive (vgl. [Broadcom 2021](#)) archiviert Protokolldaten und markiert diese in der Datenbank als archiviert.
 - AE DB Reorg (vgl. [Broadcom 2021](#)) reorganisiert Daten in der Datenbank, indem es für diese das Löschkennzeichen setzt.

2 Grundlagen

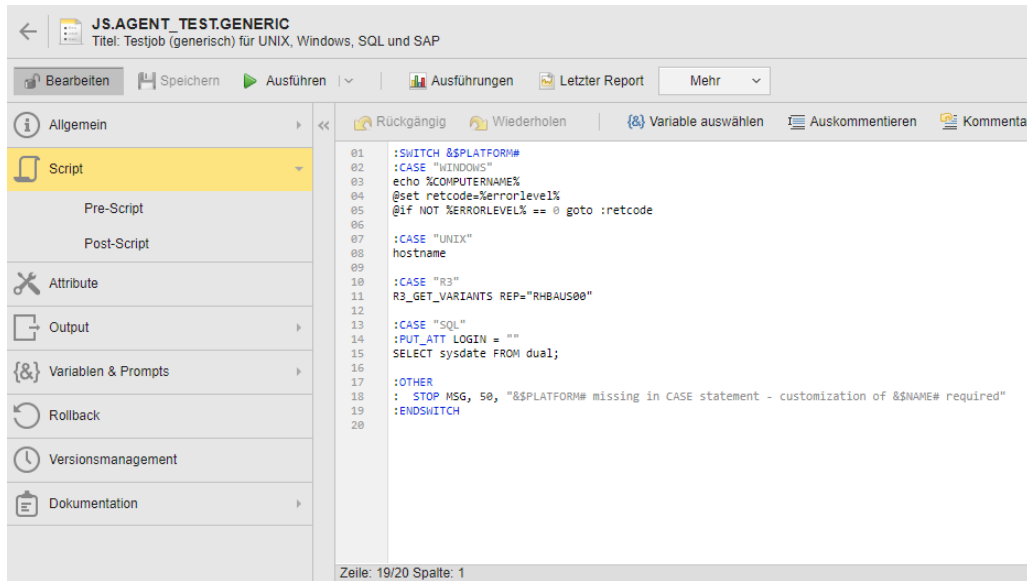


Abbildung 2.2: Job-Beispiel anhand eines generischen Jobs²

- AE DB Unload (vgl. [Broadcom 2021](#)) kann archivierte oder reorganisierte Daten aus der Datenbank entfernen.
- Archive Browser (vgl. [Broadcom 2021](#)) zum Betrachten und Durchsuchen archivierter Protokolldaten.

2.3.1 Workflows

Ein Workflow (Abb. 2.3) orchestriert die automatische Durchführung einzelner Aufgaben von beispielsweise Jobs anhand von Bedingungen und Relationen und beschreibt damit deren Abfolge (vgl. [Broadcom 2021](#)). Betrachtet man einen Workflow aus einem abstrahierten Modellierungsgrad, ist die Instanz eines Workflows selbst wieder eine Aufgabe.

Mittels Workflows können Geschäftsprozesse oder Teilprozesse aus dem BPM technisch beschrieben werden.

2.3.2 Jobs

²Quelle: Jan Thomas Gundlach, 2021 - Screenshot

2 Grundlagen

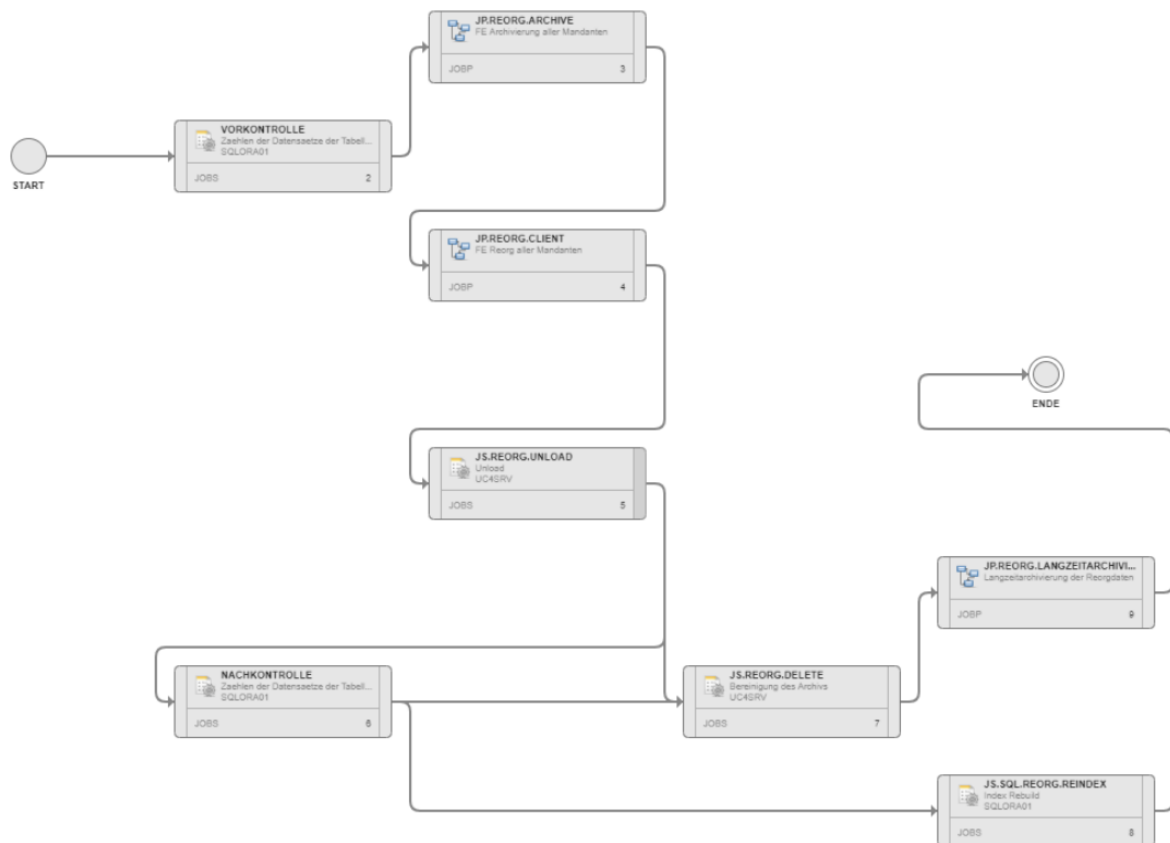


Abbildung 2.3: Beispiel eines Workflows am Reorganisationsprozess³

Ein Job (Abb. 2.2) ist eine Vorlage für eine Aufgabe, die eine oder mehrere Anweisungen in einer definierten Reihenfolge für eine durch sie spezifizierte Umgebung enthält (vgl. [Broadcom 2021](#)). Dadurch sind Jobs Basisbausteine innerhalb von automatisierten Prozessen.

Ein Job beschreibt innerhalb des BPM einen Teilprozess.

2.3.3 Ausführungen

Durch die Ausführung eines Workflows oder Jobs wird als Instanz dessen eine Aufgabe erzeugt. Die Aufgabe durchläuft anschließend während der Ausführung die Phasen Aktivierung, Generierung, Verarbeitung und Abschluss (vgl. [Ahrends 2010](#)). Eine Aufgabe beschreibt innerhalb des BPM eine Aktivität und eine Ausführung (Abb. 2.4) dessen Resultat.

³Quelle: Jan Thomas Gundlach, 2021 - Screenshot

2 Grundlagen

Name	Type	Platform	RunID	Parent	Status	Aktivierungszeit	Startzeit	Endzeit
▼ JP.KVS529A	JOBP		150914...	150279...	ENDED_OK - Normal beendet.	27.08.2021 05:00	27.08.2021 05:00	27.08.2021 07:10
▼ JP.KV...	JOBP		151005...	150914...	ENDED_OK - Normal beendet.	27.08.2021 07:17	27.08.2021 07:17	27.08.2021 07:35
▼ JS...	JOBS	Unix	151017...	151005...	ENDED_NOT_OK - abgebro...	27.08.2021 07:30	27.08.2021 07:30	27.08.2021 07:30
FR	RE...		151017...		ENDED_OK - Normal beendet.	27.08.2021 07:30	27.08.2021 07:30	27.08.2021 07:30
▶ JS...	JOBS	Unix	151005...	151005...	ENDED_INACTIVE - Aufgab...	27.08.2021 07:17		27.08.2021 07:35
▶ JS...	JOBS	Unix	151005...	151005...	ENDED_INACTIVE - Aufgab...	27.08.2021 07:17		27.08.2021 07:35
▶ JS...	JOBS	Unix	151005...	151005...	ENDED_INACTIVE - Aufgab...	27.08.2021 07:17		27.08.2021 07:35
▶ JS...	JOBS	Unix	151005...	151005...	ENDED_INACTIVE - Aufgab...	27.08.2021 07:17		27.08.2021 07:35
▶ JS...	JOBS	Unix	151005...	151005...	ENDED_INACTIVE - Aufgab...	27.08.2021 07:17		27.08.2021 07:35
▶ JS...	JOBS	Unix	151005...	151005...	ENDED_INACTIVE - Aufgab...	27.08.2021 07:17		27.08.2021 07:35
▶ JS...	JOBS	Unix	151005...	151005...	ENDED_OK - Normal beendet.	27.08.2021 07:17	27.08.2021 07:26	27.08.2021 07:26
▶ JS...	JOBS	Unix	15...	151005...	ENDED_NOT_OK - abgebro...	27.08.2021 07:17	27.08.2021 07:26	27.08.2021 07:26
▶ JS...	JOBS	Unix	151005...	151005...	ENDED_OK - Normal beendet.	27.08.2021 07:17	27.08.2021 07:26	27.08.2021 07:26
▼ JP.KV...	JOBP		150915...	150914...	ENDED_OK - Normal beendet.	27.08.2021 05:00	27.08.2021 05:00	27.08.2021 07:10
▶ JS...	JOBS	Unix	150957...	150915...	ENDED_NOT_OK - abgebro...	27.08.2021 06:05	27.08.2021 06:05	27.08.2021 06:05
▶ AL...	JOBS	Unix	150915...	150915...	ENDED_INACTIVE - Aufgab...	27.08.2021 05:00		27.08.2021 07:10
▶ AL...	JOBS	Unix	150905...	150915...	ENDED_INACTIVE - Aufgab...	27.08.2021 05:00		27.08.2021 07:10
▶ AL...	JOBS	Unix	150905...	150915...	ENDED_INACTIVE - Aufgab...	27.08.2021 05:00		27.08.2021 07:10
▶ AL...	JOBS	Unix	150915...	150915...	ENDED_INACTIVE - Aufgab...	27.08.2021 05:00		27.08.2021 07:10
▶ AL...	JOBS	Unix	150915...	150915...	ENDED_INACTIVE - Aufgab...	27.08.2021 05:00		27.08.2021 07:10
▶ JS...	JOBS	Unix	150905...	150915...	ENDED_INACTIVE - Aufgab...	27.08.2021 05:00		27.08.2021 07:10
▶ JS...	JOBS	Unix	15...	150915...	ENDED_NOT_OK - abgebro...	27.08.2021 05:00	27.08.2021 05:00	27.08.2021 05:00
▶ AL...	JOBS	Unix	150905...	150915...	ENDED_OK - Normal beendet.	27.08.2021 05:00	27.08.2021 05:00	27.08.2021 05:00
SC...	SCRI		150915...	150915...	ENDED_OK - Normal beendet.	27.08.2021 05:00	27.08.2021 05:00	27.08.2021 05:00
▶ JP.KV...	JOBP		150904...	150914...	ENDED_OK - Normal beendet.	27.08.2021 05:00	27.08.2021 05:00	27.08.2021 05:00
▶ JS.KV...	JOBS	Unix	150904...	150914...	ENDED_OK - Normal beendet.	27.08.2021 05:00	27.08.2021 05:00	27.08.2021 05:00
▼ JS.KV...	JOBS	Unix	150904...	150914...	ENDED_OK - Normal beendet.	27.08.2021 05:00	27.08.2021 05:00	27.08.2021 05:00
FR	RE...		150915...		ENDED_OK - Normal beendet.	27.08.2021 05:00	27.08.2021 05:00	27.08.2021 05:00
SC.ST...	SCRI		150914...	150914...	ENDED_OK - Normal beendet.	27.08.2021 05:00	27.08.2021 05:00	27.08.2021 05:00
▶ JP.KVS529A	JOBP		149933...	149281...	ENDED_OK - Normal beendet.	26.08.2021 05:00	26.08.2021 05:00	26.08.2021 05:01

Abbildung 2.4: Eine Liste von Ausführungen und deren Abhängigkeiten⁴

2.3.4 Reports

Beim Erzeugen einer Aufgabe werden alle Ausführungsphasen zwecks Rückverfolgbarkeit und Prüfbarkeit in einem Report protokolliert.

2.3.5 Reorganisation

Unter dem ständigen und insbesondere schnellen Wachstum einer Datenbank kann die Performanz als auch deren Wartbarkeit leiden. Parallel steigen durch den physikalischen Bedarf die Betriebskosten. Um diesem entgegen zu wirken, sollten Datenbanken regelmässig reorganisiert werden.

Bei einer Reorganisation (Abb. 2.3) werden nicht mehr benötigte oder verwaiste Daten gelöscht und bestehende Daten ggf. neu strukturiert (vgl. [Broadcom 2021](#)).

⁴Quelle: Jan Thomas Gundlach, 2021 - Screenshot

2 Grundlagen

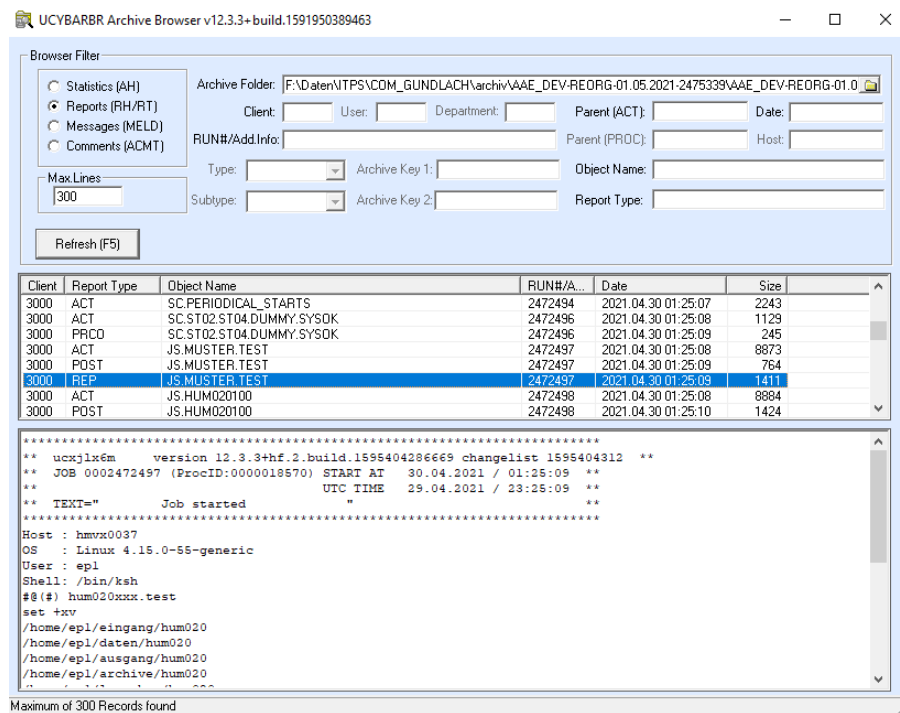


Abbildung 2.5: Ansicht archivierter Datensätze im Automatic® Archive Browser⁵

2.4 Automatic® Archive Browser

Der Automatic® Archive Browser (Abb. 2.5) ist ein Dienstprogramm, mit dem archivierte Datensätze aus Automatic® Automation zur Anzeige gebracht werden können und mit dem in diesen recherchiert werden kann (vgl. [Broadcom 2021](#)).

2.5 Rechtliche Grundlagen

Die Nachweis- und Aufbewahrungspflichten von Protokolldaten ergeben sich im Allgemeinen unter anderem durch folgende gesetzliche Regelungen:

- Datenschutz (BDSG, DSGVO)
- Handels- und Steuerrecht (HGB, AO)
- Informations- und Kommunikationsrecht (IuKDG, ITSIG)
- Vertrauensdienste und elektronische Signaturen (VDG, IVT)

⁵Quelle: Jan Thomas Gundlach, 2021 - Screenshot

- Vorschriften aus dem Bank- und Versicherungsrecht (VAG, VAIT)
- Zivil- und Strafrecht (ZPO, BGB, StGB, StPO)

Im Folgenden soll auf die hieraus sich ergebenden Auswirkungen auf die Protokolldaten aus Geschäftsprozessen innerhalb des Versicherungswesens eingegangen werden.

2.5.1 VAIT

In der Versicherungswirtschaft werden annähernd alle Geschäftsprozesse EDV-gestützt abgewickelt. Entsprechend empfindlich treffen insbesondere zentrale Störungen der IT-Systeme den Betrieb und können schnell ein geschäftskritisches Ausmaß annehmen. Beeinträchtigt werden die IT-Systeme dabei längst nicht mehr vorwiegend durch Wartungsintervalle, Entwicklungszyklen oder Kapazitätsengpässe, sondern immer mehr gezielt durch kriminelle Angriffe mit der Absicht der Sabotage oder Spionage. Letztere sind nicht nur gefährlicher, sondern in der Regel auch schwieriger zu erkennen.

Aus dem „Global Threat Report: Extended Enterprise Under Threat“ (vgl. [VMware® Carbon Black 2020](#)) geht hervor, dass zwischen Februar und April 2020 global die Angriffe auf Unternehmen um das Zweieinhalbfache zugenommen haben. Betrachtet man nur solche mit Erpressungssoftware, so findet sich sogar eine Steigerung um den Faktor Neun. In mehr als einem Viertel war die Finanzwirtschaft das Ziel der Angriffe.

Für Deutschland im Speziellen hat der „Deutschland Threat Report: Das Extended Enterprise in Gefahr“ (vgl. [VMware® Carbon Black 2020](#)) ermittelt, dass die Anzahl der Cyberangriffe von April 2019 bis April 2020 um 70% gestiegen sind. In 82 % waren die Angriffe komplexer und ausgefeilter als zuvor. 73 % der Unternehmen haben in diesem Zeitraum dadurch im Schnitt zwei Datenlecks erlitten. In einem Viertel der Vorfälle waren Schwachstellen in Prozessen die Ursache. Auf die Finanzbranche bezogen, waren Schwachstellen in Prozessen sogar in 54,5% verantwortlich, wobei es in 46% dieser Fälle zu einem Datenleck kam.

Dies verdeutlicht die IT-Sicherheitslage auch in der deutschen Versicherungswirtschaft und zeigt, wo Potenzial bei der Identifikation und Abwehr von Angriffen vorhanden ist. Laut der Studie „State of Cyber Resilience“ aus dem Jahr 2020 (vgl. [Accenture 2020](#)) wurde nicht mal jeder dritte Informationssicherheitsvorfall in der Finanzwirtschaft innerhalb von 24 Stunden erkannt. In Anbetracht der Kritikalität

von bspw. Gesundheitsdaten aus dem Krankenversicherungswesen wird die Brisanz dieser Situation ersichtlich. Hintergrund ist auch, dass Versicherer im Zuge der Digitalisierung ihrer Prozesse heterogenere und damit komplexe IT-Landschaften betreiben, was die Übersichtlichkeit mindert und die Wahrscheinlichkeit eines Vorfalles oder einer Störung erhöht.

Diesem Umstand begegnet die Bundesanstalt für Finanzdienstleistungsaufsicht (BaFin) seit Juli 2018 mit den versicherungsaufsichtlichen Anforderungen an die IT (VAIT) (vgl. [BaFin 2019](#)), um einen kontrollierten und sichereren Einsatz von IT-Umgebungen in den Versicherungsunternehmen zu erwirken. Sie fordert damit ein kleinmaschigeres IT-Risikomanagement von den Versicherern.

Die VAIT reguliert die Ausgestaltung der informationstechnischen Systeme als auch der entsprechenden Prozesse. Damit präzisiert die VAIT die rechtlichen Vorgaben des Versicherungsaufsichtsgesetzes (VAG) aus §23 bis §32. Bei diesen handelt es sich um *„normeninterpretierende Verwaltungsvorschriften, die eine Selbstbindung der deutschen Aufsicht gegenüber den Versicherungen darstellen“* (vgl. [Wikipedia 2021](#)).

Insbesondere ergeben sich aus der VAIT auch aufsichtsrechtliche Vorgaben zur Prozessgestaltung, -dokumentation und -protokollierung. Im Mittelpunkt steht hierbei die Ermittlung des unternehmensspezifischen Schutzbedarfes und die Sicherstellung der Schutzziele für denselben (3.21). Da Geschäftsprozesse durch IT-Systeme und Prozesse unterstützt werden (2.8), muss sichergestellt und regelmäßig überprüft werden, dass die ermittelten IT-Prozesse geeignet sind, die Schutzziele zu erreichen (2.15).

Als Grundlage muss eine Informationssicherheitsleitlinie definiert werden, welche in Informationssicherheitsrichtlinien für die jeweiligen Prozesse, insbesondere im Bereich Protokollierung, umgesetzt wird (4.27). In der Folge wird insbesondere für den ermittelten Schutzbedarf eine Prozess- und Änderungsdokumentation verlangt (2.7). Parallel sollte es Prozesse zur Protokollierung und Überwachung geben (5.40), die mitunter Systemaktivitäten diesbezüglich protokollieren (6.51). Die Implementierung der Protokollierung muss manipulationssicher erfolgen (5.41). Damit verbunden ist die Erwartung, dass neben anderen Aufgaben zu jeder Zeit ein Überblick über die Geschäftsprozesse sowie deren Abhängigkeiten und Schnittstellen zur Verfügung steht (3.20).

Sollte es dennoch zu Informationssicherheitsvorfällen (4.31) oder Störungen (7.63) betroffener Geschäftsprozesse kommen, hat eine entsprechende Analyse in Korrelation

zur Ursache und Nachverfolgung mit angemessenen Nachsorgemaßnahmen zu erfolgen. Im Vorwege müssen für Teilprozesse hiervon Berichtspflichten definiert werden (3.17).

Für die Bereiche der kritischen Infrastruktur (KRITIS) muss der Geltungsbereich gemäß der Anlage der Verordnung zur Bestimmung Kritischer Infrastrukturen nach dem BSI-Gesetz (BSI-KritisV) vollständig nachgewiesen werden können (9.72).

Im Folgenden die verwiesenen Auszüge aus der VAIT dazu:

VAIT Abschnitt 2. IT-Governance Punkt 7:

„[...]

Prozesse sowie die damit verbundenen Aufgaben, Kompetenzen, Verantwortlichkeiten, Kontrollen sowie Kommunikationswege sind klar zu definieren und aufeinander abzustimmen.

[...] “ (BaFin 2019: 8)

VAIT Abschnitt 2. IT-Governance Punkt 8:

„[...]

Die Informationsverarbeitung und -weitergabe in Geschäfts- und Serviceprozessen wird durch datenverarbeitende IT-Systeme und zugehörige IT-Prozesse unterstützt.

[...] “ (BaFin 2019: 8)

VAIT Abschnitt 2. IT-Governance Punkt 15:

„Die IT-Systeme und die zugehörigen IT-Prozesse müssen die Integrität, die Verfügbarkeit, die Authentizität sowie die Vertraulichkeit der Daten sicherstellen. ... Die Eignung der IT-Systeme und der zugehörigen IT-Prozesse, die Schutzziele zu erreichen, ist regelmäßig von den fachlich und technisch zuständigen Mitarbeitern zu überprüfen.“ (BaFin 2019: 9)

VAIT Abschnitt 3. Informationsrisikomanagement Punkt 17:

„... Das Unternehmen hat angemessene Identifikations-, Bewertungs-, Überwachungs- und Steuerungsprozesse einzurichten und diesbezügliche Berichtspflichten zu definieren.“ (BaFin 2019: 11)

VAIT Abschnitt 3. Informationsrisikomanagement Punkt 20:

„Das Unternehmen hat über einen aktuellen Überblick über die Bestandteile des festgelegten Informationsverbunds sowie deren Abhängigkeiten und Schnittstellen zu verfügen.

Zu einem Informationsverbund gehören beispielsweise geschäftsrelevante Informationen, Geschäftsprozesse, IT-Systeme ... “ (BaFin 2019: 11)

VAIT Abschnitt 3. Informationsrisikomanagement Punkt 21:

„... Ermittlung des Schutzbedarfs (insbesondere im Hinblick auf die Schutzziele „Integrität“, „Verfügbarkeit“, „Vertraulichkeit“ und „Authentizität“) ... “ (BaFin 2019: 11)

VAIT Abschnitt 4. Informationssicherheitsmanagement Punkt 27:

„... Informationssicherheitsrichtlinien und Informationssicherheitsprozesse mit den Teilprozessen Identifizierung, Schutz, Entdeckung, Reaktion und Wiederherstellung zu definieren.

Informationssicherheitsrichtlinien werden beispielsweise für die Bereiche Netzwerksicherheit, Kryptografie, Authentisierung und Protokollierung erstellt.

[...] “ (BaFin 2019: 13)

VAIT Abschnitt 4. Informationssicherheitsmanagement Punkt 31:

„Nach einem Informationssicherheitsvorfall sind die Auswirkungen auf die Informationssicherheit zu analysieren und angemessene Nachsorgemaßnahmen zu veranlassen.

Die Definition des Begriffes „Informationssicherheitsvorfall“ nach Art und Umfang basiert auf dem Schutzbedarf der betroffenen Geschäftsprozesse, IT-Systeme und den zugehörigen IT-Prozessen.

[...] “ (BaFin 2019: 15)

VAIT Abschnitt 5. Benutzerberechtigungsmanagement Punkt 40:

„Das Unternehmen hat nach Maßgabe des Schutzbedarfs und der Soll-Anforderungen Prozesse zur Protokollierung und Überwachung einzurichten ... “ (BaFin 2019: 17)

VAIT Abschnitt 5. Benutzerberechtigungsmanagement Punkt 41:

*„Technisch-organisatorische Maßnahmen hierzu sind beispielsweise:
[...]*

- *eine manipulationssichere Implementierung der Protokollierung,*

[...] “ (BaFin 2019: 17)

VAIT Abschnitt 6. IT-Projekte, Anwendungsentwicklung (inkl. durch Endbenutzer in den Fachbereichen) Punkt 51:

„Im Rahmen der Anwendungsentwicklung sind nach Maßgabe des Schutzbedarfs angemessene Vorkehrungen im Hinblick darauf zu treffen, dass nach Produktivsetzung der Anwendung die Vertraulichkeit, Integrität, Verfügbarkeit und Authentizität der zu verarbeitenden Daten nachvollziehbar sichergestellt werden.

Geeignete Vorkehrungen können sein:

- *Prüfung der Eingabedaten,*
- *Systemzugangskontrolle,*
- *Nutzer-Authentifizierung,*
- *Transaktionsautorisierung,*
- *Protokollierung der Systemaktivität,*
- *Prüfpfade (Audit Logs),*
- *Verfolgung von sicherheitsrelevanten Ereignissen,*
- *Behandlung von Ausnahmen.“ (BaFin 2019: 19-20)*

VAIT Abschnitt 7. IT-Betrieb (inkl. Datensicherung) Punkt 63:

„Die Meldungen über ungeplante Abweichungen vom Regelbetrieb (Störungen) und deren Ursachen sind in geeigneter Weise zu erfassen ... Bearbeitung, Ursachenanalyse und Lösungsfindung inkl. Nachverfolgung sind zu dokumentieren. Ein geordneter Prozess zur Analyse möglicher Korrelationen von Störungen und deren Ursachen muss vorhanden sein.

[...] “ (BaFin 2019: 23)

VAIT Abschnitt 9. Kritische Infrastrukturen Punkt 72:

„Der Geltungsbereich für die Nachweiserbringung für die kritische Infrastruktur muss die Anlage gemäß BSI-KritisV vollständig umfassen.

[...] “ (BaFin 2019: 27)

Für die Erfüllung dieser Anforderungen ist der Einsatz einer Process Engine eine Möglichkeit. Zwar gibt es seitens der BaFin bisher keine Meldepflichten für den gesamten Versicherungssektor (vgl. BaFin 2020). Zum Nachweis der Einhaltung der Anforderungen können jedoch Erkenntnisse mittels der Meldepflichten aus dem Anzeige- und Meldewesen herangezogen werden. Hieraus lassen sich bspw. gezielt Sonderprüfungen ableiten. Grundsätzlich können Prüfungen sowohl aus besonderem Grund wie auch als Routineprüfungen ohne einen besonderen Grund durchgeführt werden.

Unabhängig vom Hintergrund ist die Einhaltung der Anforderungen in einer Prüfung ggf. auch anhand von Stichproben nachzuweisen. In der Folge müssen diese innerhalb der Aktivitäten und deren Abhängigkeiten in der Process Engine auch historisch erfüllt und nachgewiesen werden können.

Zugleich sind die VAIT ein passender Leitfaden für die Prüfung der internen Revision, da sie das aufsichtsrechtliche Feld an die Informationstechnik eines Versicherungsunternehmens vorgeben (vgl. Q_PERIOR AG 2019). Die Vorgaben haben daher auch Auswirkungen auf den Prüfungsansatz und die inhaltliche Gestaltung der Prüfungen der internen Revision. Besonders die Prüfung schnittstellenübergreifender Prozesse steht im Bezug zur VAIT meist im Fokus der internen Revision und damit auch die Relationen dazugehöriger Teilprozesse.

2.5.2 ITSiG / BSI-KritisV

Das IT-Sicherheitsgesetz (ITSiG) verpflichtet Unternehmen und Behörden dazu, Maßnahmen durchzuführen, um die Sicherheit informationstechnischer Systeme zu gewährleisten und nachzuweisen. Außerdem soll der Schutz der Bürgerinnen und Bürger der Bundesrepublik Deutschland im Internet durch das ITSiG verbessert werden. Das ITSiG nimmt mit einzelnen Regelungen Einfluss auf Betreiber von kommerziellen Webangeboten und Telekommunikationsunternehmen. Neben Gesetzen wie dem Energiewirtschaftsgesetz, Atomgesetz, Telemediengesetz und dem Telekommunikationsgesetz hat das ITSiG auch das Gesetz über das Bundesamt für Sicherheit in der Informationstechnik (BSiG) erweitert. Primär ist das Gesetz zum Schutz vor Störungen in der KRITIS verabschiedet worden. Die vom ITSiG abgeleitete BSI-KritisV definiert, wer Betreiber von KRITIS ist.

Die Schutzziele des ITSiG sind im Kern:

- Die **Verfügbarkeit** von informationstechnischen Systemen und Komponenten sicherzustellen.
- Die **Integrität** von übertragenen Daten zu gewährleisten.
- Die **Vertraulichkeit** von schützenswerten Daten zu garantieren.
- Die **Authentizität** der Identität von Kommunikationspartnern festzustellen.

Laut Bundesamt für Sicherheit in der Informationstechnik (BSI) müssen die Betreiber Kritischer Infrastrukturen:

„gemäß § 8a Absatz 1 BSiG ihre Vorkehrungen zur Vermeidung von Störungen der Verfügbarkeit, Integrität, Authentizität und Vertraulichkeit ihrer informationstechnischen Systeme, Komponenten oder Prozesse, die für die Funktionsfähigkeit der von ihnen betriebenen Kritischen Infrastrukturen maßgeblich sind, gegenüber dem BSI auf geeignete Weise nachweisen.“
(vgl. [BSI 2021](#))

Laut BSI §8a Absatz 3 muss der Nachweis alle zwei Jahre durch ein KRITIS-Audit (Sicherheitsaudit, Prüfung oder Zertifizierung) erbracht werden. Die Prozessdokumentation ist unter anderem ein regulatorisches Erfordernis und Grundlage der Au-

ditierung. Der Auditor kann sich darüber hinaus Protokolle von Prozessen und deren Abhängigkeiten zeigen lassen, um beispielsweise die Einhaltung der Prozessdokumentation zu überprüfen.

2.5.3 DSGVO

Mit der Datenschutz-Grundverordnung (DSGVO) wird der rechtliche Rahmen zur Verarbeitung von personenbezogenen Daten durch öffentliche Einrichtungen und Unternehmen im europäischen Binnenmarkt harmonisiert (vgl. [intersoft consulting services AG 2021](#)). Sie stärkt die Rechte der Verbraucher durch informationelle Selbstbestimmung. Dazu legt sie den Verantwortlichen umfangreiche Vorschriften für die Erhebung und Verarbeitung von Daten auf. Diese müssen seit dem 25. Mai 2018 beispielsweise im Ablauf von Geschäftsprozesse berücksichtigt werden, damit diese DSGVO-konform betrieben werden. Somit modernisiert die DSGVO die EU-Datenschutzrichtlinie in der Europäischen Union hinsichtlich einer fortschreitenden Digitalisierung und daraus ableitender Datenverarbeitung (vgl. [Bundesministerium für Wirtschaft und Energie 2021](#)).

Besonders relevant für diese Arbeit sind die Rechtsgrundlagen für Datenverarbeitungen der DSGVO. Die DSGVO findet Anwendung, sobald eine automatisierte Verarbeitung personenbezogener Daten durchgeführt wird (Art. 2 Abs. 1 DSGVO) ([Amtsblatt der Europäischen Union 2016](#): L 119/32). Auch bei einer nichtautomatisierten Verarbeitung gilt die DSGVO, wenn personenbezogene Daten in einem Dateisystem gespeichert sind oder gespeichert werden (Art. 2 Abs. 1 DSGVO) ([Amtsblatt der Europäischen Union 2016](#): L 119/32). Für die Verarbeitung lassen sich dabei aus Art. 5 Abs. 1 DSGVO ([Amtsblatt der Europäischen Union 2016](#): L 119/35-36) unmittelbar verschiedene Anforderungen u. a. Transparenz, Zweckbindung, Richtigkeit, Integrität und Vertraulichkeit ableiten ([Datenschutzkonferenz 2019](#): 13). Parallel verlangt die DSGVO von einem Unternehmen, „geeignete technische und organisatorische Maßnahmen“ zu ergreifen, um diese Schutzziele der DSGVO zu erreichen. In Art. 32 Abs. 1 lit. b DSGVO ([Amtsblatt der Europäischen Union 2016](#): L 119/52) werden als solche Maßnahmen explizit *„Vertraulichkeit, Integrität, Verfügbarkeit und Belastbarkeit der Systeme“* und in Art. 32 Abs. 1 lit. d DSGVO ([Amtsblatt der Europäischen Union 2016](#): L 119/52) *„regelmäßigen Überprüfung, Bewertung und Evaluierung der Wirksamkeit der technischen und organisatorischen Maßnahmen“* benannt.

In der DSGVO sind für die Verarbeitung als auch die technischen und organisatorischen Maßnahmen umfassende Rechenschaftspflichten verankert. Von dem verantwortlichen Unternehmen wird in Art. 5 Abs. 2 DSGVO ([Amtsblatt der Europäischen Union 2016](#):L 119/36) gefordert, die konforme Verarbeitung im Zweifel nachweisen zu können. Konkretisiert wird dies, indem das Unternehmen nach Art. 24 Abs. 1 DSGVO ([Amtsblatt der Europäischen Union 2016](#):L 119/47) geeignete technische und organisatorische Maßnahmen umzusetzen hat, um den Nachweis sicherzustellen, dass die Verarbeitung gemäß DSGVO erfolgt (vgl. [Niklas Plutte 2021](#)).

Die DSGVO definiert nicht, wie lange ein solcher Nachweis rückblickend geführt werden muss. Für Bußgelder nach Art. 83 DSGVO ([Amtsblatt der Europäischen Union 2016](#):L 119/82) wegen Verstößen gegen die DSGVO gelten nach § 41 Bundesdatenschutzgesetz die Vorschriften und somit auch dessen Verjährungsvorschriften, des Gesetzes über Ordnungswidrigkeiten. In diesem ist in § 31 die Verfolgungsverjährung bzgl. Bußgeldverfahren gegenüber Aufsichtsbehörden festgeschrieben, nach der ein Bußgeld nach Art. 83 DSGVO drei Jahre nach einer vorschriftswidrigen Verarbeitung verjährt ist (vgl. [Christina Webersohn 2019](#)). Entsprechend sollten Nachweise mind. 3 Jahre aufbewahrt werden, da ein weiter zurückliegender Verstoß risikolos wäre. Dies ist ein Zeitraum, in dem zurückliegende Daten innerhalb einer Process Engine in der Regel teilweise bis vollständig reorganisiert und archiviert werden.

2.6 Java Implementierung

Eine Java Implementierung ist eine Implementierung der Java Virtual Machine (JVM) auf einer spezifischen Java Plattform nach der Java Language Specification (JLS) (vgl. [Oracle 2021](#)). Die JLS wird im Java Community Process (JCP) von Java Specification Request (JSR) weiterentwickelt (vgl. [Oracle Corporation 2021](#)). Hierbei definiert ein angenommener und abschließend bestätigter JSR einen neuen Java-Standard (vgl. [tutego 2011](#)). Zugleich stellt dieser eine Referenzimplementierung des neuen Java-Standards und ein Technology Compatibility Kit (TCK) hierfür zur Verfügung. Mit einem TCK kann eine Java Implementierung gegen den entsprechenden Sprachstandard validiert werden und darf nach Zertifizierung anschließend als Java SE kompatibel bezeichnet werden (vgl. [Julius Stiebert 2007](#)). Eine Java Implementierung wird in der Regel als Java Development Kit (JDK) veröffentlicht.

2.7 GUI-Framework

Ein Graphical User Interface (GUI) Framework (vgl. [Siddiqi 2020](#)) stellt mittels einer Bibliothek typische Steuerelemente zur Erstellung von grafischen Benutzeroberflächen zur Verfügung, über die ein Endanwender mit einer dazugehörigen Anwendung interagieren kann. Das Framework kapselt dabei primitive, grafische Objekte des Betriebssystems bis hin zu eigenen grafischen Elementen in geeignete Strukturen bzw. Klassen. Es kann außerdem diese um Funktionalität erweitern und sie als Bedienelemente zur Verfügung stellen. Für letztere verfügt das Framework in der Regel über eine eigene Render-Engine. Mittels des Frameworks können so innerhalb integrierter Entwicklungsumgebungen oder auch unabhängig davon einfache grafische Benutzeroberflächen geschaffen werden, da die Steuerelemente oder Widgets (vgl. [Informit 2005](#)) hierzu nicht mehr eigenständig entwickelt werden müssen.

Viele GUI Frameworks verfügen darüber hinaus über einen Designer zur mausgesteuerten Erstellung sowie einen Layoutmanager zur korrekten Anordnung der Steuerelemente.

3 Analyse

3.1 Automic® Archive Browser

3.1.1 Ermittlung von Benutzeranforderungen

Zur Umsetzung eines neuen Recherchewerkzeuges für Archivdaten müssen die Erwartungen und Bedürfnisse der Benutzer geklärt werden.

Funktionsumfang

In einem ersten Schritt wurde der bestehende Funktionsumfang (Tab. 3.1) ermittelt und als Grundlage für einen Workshop mit den Benutzern dokumentiert.

Definition Automic® Archive Browser Funktionsumfang				
Funktion	Ausführungsdaten	Protokolle	Meldungen	Kommentare
Mandanten laden	X	X	X	X
Mandanten laden mit mehreren Archiven	X	X	X	X
Mehrere Mandanten laden	X	X	X	X
Mehrere Mandanten laden mit mehreren Archiven	-	-	-	-
Begrenzung Max.Lines	X	X	X	X
Anzeige nur Ausführungsdaten (AH)	X	-	-	-
Anzeige nur Protokolle (RH/RT)	-	X	-	-
Anzeige nur Meldungen (MELD)	-	-	X	-
Anzeige nur Kommentare (ACMT)	-	-	-	X
Ausführungsdaten, Protokolle, Kommentare kombiniert	-	-	-	-
Ausführungsdaten, Protokolle kombiniert	-	-	-	-
Relationen (RUN#, Parent (ACT))	-	-	-	-

Fortsetzung auf der nächsten Seite

3 Analyse

Merken der markierten Zeile beim Suchen	-	-	-	-
Mitscrollen der markierten Zeile beim Suchen	-	-	-	-
Spalte Client	X	X	X	X
Spalte User	X	-	X	X
Spalte Department	X	-	X	X
Spalte Archive Key 2	X	-	-	-
Spalte Object Type	X	-	-	-
Spalte RUN#/Add.Info	X	X	-	X
Spalte Parent (ACT)	X	-	-	-
Spalte Parent (PROC)	X	-	-	-
Spalte Date	X	X	X	X
Spalte Host	X	-	X	-
Spalte Subtype	X	-	-	-
Spalte Archive Key 1	X	-	-	-
Spalte Object Name	X	X	-	X
Spalte Report Type	-	X	-	-
Spalte Size	-	X	-	-
Spalte Source	-	-	X	-
Spalte Category	-	-	X	-
Spalte Type	-	-	X	-
Spalte Status	-	-	-	-
Mehrfach sortieren	X	X	X	X
Suche Client	X	X	X	X
Suche User	X	-	X	X
Suche Department	X	-	X	X
Suche Parent (ACT)	X	-	-	-
Suche Date	X	X	X	X
Suche RUN#/Add.Info	X	X	-	-
Suche Parent (PROC)	X	-	-	-
Suche Host	X		X	X
Suche Type	X	-	-	-
Suche Archive Key 1	X	-	-	-
Suche Object Name	X	X	-	-
Suche Message Text	-	-	X	X

Fortsetzung auf der nächsten Seite

3 Analyse

Suche Subtype	X	-	-	-
Suche Archive Key 2	X	-	-	-
Suche Report Type	-	X	-	-
Kombinierte Suche	X	X	X	X
Übergreifende Suche	-	-	-	-

Tabelle 3.1: Funktionsumfang des Automic® Archive Browser v12.3.3+build1591950389463 (Abb. 2.5)¹

Benutzeranforderungen

In einem WebEx-Workshop vom 27.07.2021 wurde die Idee eines relationsbasierten Recherchewerkzeuges vorgestellt und auf Basis des bestehenden Funktionsumfangs daran geknüpfte Anforderungen der Benutzer definiert.

Der Workshop hat für ein relationsbasiertes Recherchewerkzeug zu den Benutzeranforderung unter Tab. 3.2 geführt.

Benutzeranforderungen relationsbasiertes Archivdaten-Recherchewerkzeug				
Funktion	Ausführungsdaten	Protokolle	Meldungen	Kommentare
Mandanten laden	X	X	-	X
Mandanten laden mit mehreren Archiven	X	X	-	X
Mehrere Mandanten laden	X	X	-	X
Mehrere Mandanten laden mit mehreren Archiven	X	X	-	X
Begrenzung Max.Lines	-	-	-	-
Anzeige nur Ausführungsdaten (AH)	-	-	-	-
Anzeige nur Protokolle (RH/RT)	-	-	-	-
Anzeige nur Meldungen (MELD)	-	-	-	-
Anzeige nur Kommentare (ACMT)	-	-	-	-
Ausführungsdaten, Protokolle, Kommentare kombiniert	X	X	-	X
Ausführungsdaten, Protokolle kombiniert	-	-	-	-
Relationen (RUN#, Parent (ACT))	X	X	-	X

Fortsetzung auf der nächsten Seite

¹Quelle: Jan Thomas Gundlach, 2021 - Analyse Automic® Archive Browser v12.3.3+build1591950389463

3 Analyse

Merken der markierten Zeile beim Suchen	X	X	-	X
Mitscrollen der markierten Zeile beim Suchen	X	X	-	X
Spalte Client	X	X	-	X
Spalte User	-	-	-	-
Spalte Department	-	-	-	-
Spalte Archive Key 2	X	X	-	X
Spalte Object Type	X	X	-	X
Spalte RUN#/Add.Info	X	X	-	X
Spalte Parent (ACT)	X	X	-	X
Spalte Parent (PROC)	-	-	-	-
Spalte Date	X	X	-	X
Spalte Host	X	X	-	X
Spalte Subtype	-	-	-	-
Spalte Archive Key 1	X	X	-	X
Spalte Object Name	X	X	-	X
Spalte Report Type	-	-	-	-
Spalte Size	-	-	-	-
Spalte Source	-	-	-	-
Spalte Category	-	-	-	-
Spalte Type	-	-	-	-
Spalte Status	-	-	-	-
Mehrfach sortieren	X	X	-	X
Suche Client	X	X	-	X
Suche User	-	-	-	-
Suche Department	-	-	-	-
Suche Parent (ACT)	X	X	-	X
Suche Date	X	X	-	X
Suche RUN#/Add.Info	X	X	-	X
Suche Parent (PROC)	-	-	-	-
Suche Host	X	X	-	X
Suche Type	X	X	-	X
Suche Archive Key 1	X	X	-	X
Suche Object Name	X	X	-	X
Suche Message Text	-	-	-	-

Fortsetzung auf der nächsten Seite

3 Analyse

Suche Subtype	-	-	-	-
Suche Archive Key 2	X	X	-	X
Suche Report Type	-	-	-	-
Kombinierte Suche	X	X	-	X
Übergreifende Suche	X	X	-	X

Tabelle 3.2: Benutzeranforderungen für ein relationsbasiertes Archivdaten-Rechercheinstrument für Automic® Automation²

Funktionsumfang Implementierung

Auf Grundlage der Benutzeranforderung (Tab. 3.2) wurde der zu implementierende Funktionsumfang definiert.

Innerhalb der Analyse der Abhängigkeiten zwischen den Archivdaten ist ein Softwarefehler im AE DB Archiv (*ucydbbar*) aufgefallen. Durch diesen werden ungültige RUNIDs in die Archivdaten für Kommentare (*UC_ACMT.TXT*) geschrieben. Obwohl Kommentare über die RunID mit den anderen Archivdaten in Relation gesetzt werden könnten, ist innerhalb der Analyse ein Produktfehler im Dienstprogramm AE DB Archive festgestellt worden der dies unterbindet. Daher ist die Benutzeranforderung an dieser Stelle innerhalb dieser Arbeit nicht erfüllbar und wurde für den zu implementierenden Funktionsumfang in dieser Arbeit zurückgestellt. Der Softwarefehler wurde beim TIER1-Partner des Herstellers (vgl. [Capture 2021](#)) gemeldet.

Da die Meldungen von Ihrer Datenstruktur grundsätzlich relationslos in Bezug auf Ausführungen abgelegt werden, wurden sie durch die Benutzer als nicht relevant definiert und aus der Implementierung ausgeschlossen.

Der zu implementierende Funktionsumfang für ein relationsbasiertes Archivdaten-Rechercheinstrument ist in Tab. 3.3 beschrieben.

²Quelle: Jan Thomas Gundlach, 2021 - Workshop Relationsbasiertes Archivdaten Rechercheinstrument vom 27.07.2021

3 Analyse

Funktionsumfang Implementierung relationsbasiertes Archivdaten-Recherche Werkzeug

Funktion	Ausführungs- daten	Protokolle	Meldungen	Kommentare
Mandanten laden	X	X	-	-
Mandanten laden mit mehreren Archiven	X	X	-	-
Mehrere Mandanten laden	X	X	-	-
Mehrere Mandanten laden mit mehreren Archiven	X	X	-	-
Begrenzung Max.Lines	-	-	-	-
Anzeige nur Ausführungsdaten (AH)	-	-	-	-
Anzeige nur Protokolle (RH/RT)	-	-	-	-
Anzeige nur Meldungen (MELD)	-	-	-	-
Anzeige nur Kommentare (ACMT)	-	-	-	-
Ausführungsdaten, Protokolle, Kommentare kombiniert	-	-	-	-
Ausführungsdaten, Protokolle kombiniert	X	X	-	-
Relationen (RUN#, Parent (ACT))	X	X	-	-
Merken der markierten Zeile beim Suchen	X	X	-	-
Mitscrollen der markierten Zeile beim Suchen	X	X	-	-
Spalte Client	X	X	-	-
Spalte User	-	-	-	-
Spalte Department	-	-	-	-
Spalte Archive Key 2	X	X	-	-
Spalte Object Type	X	X	-	-
Spalte RUN#/Add.Info	X	X	-	-
Spalte Parent (ACT)	X	X	-	-
Spalte Parent (PROC)	-	-	-	-
Spalte Date	X	X	-	-
Spalte Host	X	X	-	-
Spalte Subtype	-	-	-	-
Spalte Archive Key 1	X	X	-	-
Spalte Object Name	X	X	-	-
Spalte Report Type	-	-	-	-
Spalte Size	-	-	-	-
Spalte Source	-	-	-	-

Fortsetzung auf der nächsten Seite

3 Analyse

Spalte Category	-	-	-	-
Spalte Type	-	-	-	-
Spalte Status	-	-	-	-
Mehrfach sortieren	X	X	-	-
Suche Client	X	X	-	-
Suche User	-	-	-	-
Suche Department	-	-	-	-
Suche Parent (ACT)	X	X	-	-
Suche Date	X	X	-	-
Suche RUN#/Add.Info	X	X	-	-
Suche Parent (PROC)	-	-	-	-
Suche Host	X	X	-	-
Suche Type	X	X	-	-
Suche Archive Key 1	X	X	-	-
Suche Object Name	X	X	-	-
Suche Message Text	-	-	-	-
Suche Subtype	-	-	-	-
Suche Archive Key 2	X	X	-	-
Suche Report Type	-	-	-	-
Kombinierte Suche	X	X	-	-
Übergreifende Suche	X	X	-	-

Tabelle 3.3: Zu implementierender Funktionsumfang für ein relationsbasiertes Archivdaten-Rechercheinstrument für Automic® Automation³

3.1.2 Weiterentwicklung

Vor einer Neuentwicklung wurde die Weiterentwicklung der bestehenden Lösung betrachtet. In der Auslieferung des Automic® Automation Image gab es bis zu der Version 11.2.6 den Ordner TOOLS\SOURCE\UCYBARBR mit dem Quellcode des Automic® Archive Browser (vgl. [UC4 Software GmbH 2013](#)) in Microsoft Visual Basic (VB).

An der Struktur des Projektes (vgl. [MarshallSoft Computing, Inc. 2017](#)) und der

³Quelle: Jan Thomas Gundlach, 2021

Format-Spezifikation im Quellcode (VERSION 5.0) ist zu erkennen, dass es sich um ein VB 5.0 Projekt handelt, siehe Auflistung 3.1.

Listing 3.1: Automic® Archive Browser Microsoft VB 5.0 Projekt

Verzeichnis von TOOLS\SOURCE\UCYBARBR

25.08.2017	14:15	<DIR>	.
25.08.2017	14:15	<DIR>	..
21.10.2015	23:14		28.226 Browse.bas
21.10.2015	23:14		43.503 Browse.frm
21.10.2015	23:14		300.458 Browse.frx
21.10.2015	23:14		10.429 browsef.ctl
21.10.2015	23:14		330 browsef.ctx
21.10.2015	23:14		7.199 frmFind.frm
21.10.2015	23:14		32 frmFind.frx
21.10.2015	23:14		5.353 TooBig.frm
21.10.2015	23:14		2.250 TooBig.frx
21.10.2015	23:14		1.263 UCYBARbr.vbp
		10 Datei(en),	399.043 Bytes
		2 Verzeichnis(se)	

Die VB 5.0 Version basiert noch nicht auf dem .NET-Framework, sondern ist rein objektbasiert (vgl. [ComputerWeekly.de 2016](#)). Entsprechend stehen keine Sprachmittel aus dem .NET-Framework zur Verfügung. Für den Nachfolger des auf dem .NET-Framework basierenden VB .NET (vgl. [Thomas Theis 2017](#)) hat Microsoft den Sprachumfang des .NET-Frameworks ab 2017 eingeschränkt. Damals änderte Microsoft seine Strategie bei der Weiterentwicklung der Programmiersprachen VB und C# (vgl. [Microsoft 2017](#)). Davor war die Strategie, neue Features in beiden Sprachen zur Verfügung zu stellen. Seit dem Strategiewechsel gibt es eine Abstufung zwischen den beiden Sprachen. Neue .NET Features sind seitdem ggf. nicht mehr in VB zu finden. 2020 hat Microsoft bekanntgegeben, VB .NET als Sprache nicht mehr weiterzuentwickeln (vgl. [Microsoft 2020](#)). Von da an werden unter Umständen neue Funktionen aus dem .NET-Framework, die Änderungen an VB .NET als Programmiersprache notwendig machen würden, von VB .NET nicht mehr unterstützt.

Gefürchtetste Programmiersprache 2018	
Programmiersprache	Unbeliebt
Visual Basic 6	89.9%
Cobol	84.1%
CoffeeScript	82.7%
VB.NET	80.9%
VBA	80.0%
Matlab	77.4%
Assembly	71.4%
Perl	71.3%
Objective-C	70.3%
Lua	68.2%
Groovy	66.4%
Delphi/Object Pascal	65.1%
C	62.6%
Ocaml	58.5%
PHP	58.4%
Hack	57.9%
C++	53.3%
Erlang	52.8%
Ruby	52.6%
R	50.6%
Java	49.3%
Julia	47.2%
Haskell	46.4%
CSS	44.9%
HTML	44.3%

Tabelle 3.4: Liste unbeliebter Programmiersprachen bei Entwicklern⁴

Gefördert wurde diese Entwicklung durch Nutzer anderer Hochsprachen wie Java, C oder C++, die eine syntaktische Nähe zu C# und Distanz zu VB haben. Zusätzlich gilt VB in dieser Gemeinschaft als einfache Einstiegssprache mit einem schlechten Ruf (vgl. Heise-Medien 2020). Dass die Ökosysteme, die verbreitete Syntax und der Umfang von Sprachen wie Java oder C# inzwischen VB weitaus überlegen sind, wird am Stack Overflow Developer Survey von 2018 (Tab. 3.4) deutlich. In diesem führte Visual Basic die Liste der Sprachen an, die Nutzer ungern einsetzen. Zugleich wurde

⁴Quelle: Stackoverflow Developer Survey Results 2018 https://insights.stackoverflow.com/survey/2018#technology_-_most-loved-dreaded-and-wanted-languages Aufruf: 28.07.2021

VB dort 2018 das letzte Mal aufgeführt. Seit 2019 ist VB nicht mehr Bestandteil der Erhebung des Stack Overflow Developer Survey. Mittlerweile führt der nahe Verwandte Visual Basic for Applications (VBA) die Rangliste (vgl. [Stack Exchange Inc 2021](#)) nach COBOL an.

Entsprechend ungeeignet ist VB im Jahr 2021 als Grundlage für ein Softwareprojekt mit gängigen Anforderungen an Portierbarkeit, Erweiterbarkeit und Wartbarkeit.

Erschwerend kommt hinzu, dass der Quellcode des Automic® Archive Browser in der Auslieferung des aktuellen Automic® Automation Image 12.3.6⁵ nicht mehr enthalten ist. Auch wenn der Automic® Archive Browser nur noch gepflegt und nicht mehr weiterentwickelt wird (vgl. [UC4 Software GmbH 2013](#)), kann es durch die Pflege zu Quellcodeanpassungen gekommen sein, um die der öffentliche Quellcode zusätzlich erweitert werden müsste.

Eine Portierung des bestehenden Automic® Archive Browser VB 5.0 Projektes auf .NET Core würde eine Umstellung auf neue Technologien notwendig machen (vgl. [Dev-Insider 2020](#)). Der Aufwand hierzu, zusätzlich zur Erweiterung, wird als höher im Vergleich zu einer Neuentwicklung eingeschätzt.

3.1.3 Datenstruktur

Die archivierten Daten von Ausführungen und Protokollen werden durch das Dienstprogramm AE DB Archive (vgl. 2.3) in einer Ordnerstruktur (Lst. 3.2) mit 3 Ebenen abgelegt.

Listing 3.2: Ordnerstruktur von archivierten Daten

Struktur:

<ARCHIVORDNER>/<MANDANT>/<ARCHIVIERUNGSLAUF>

Beispiel:

AAE_DEV-REORG-01.05/0004/UC_ARCHIV_20210501_55

Die regulär in ISO 8859-1 kodierten, archivierten Ausführungsdaten und Protokolle unterteilen sich je Tag, Mandant und Archivierungslauf in sechs Dateien (Lst. 3.3).

⁵Quelle: Automic® Download Center <https://docs.automic.com/downloads> Aufruf: 03.08.2021

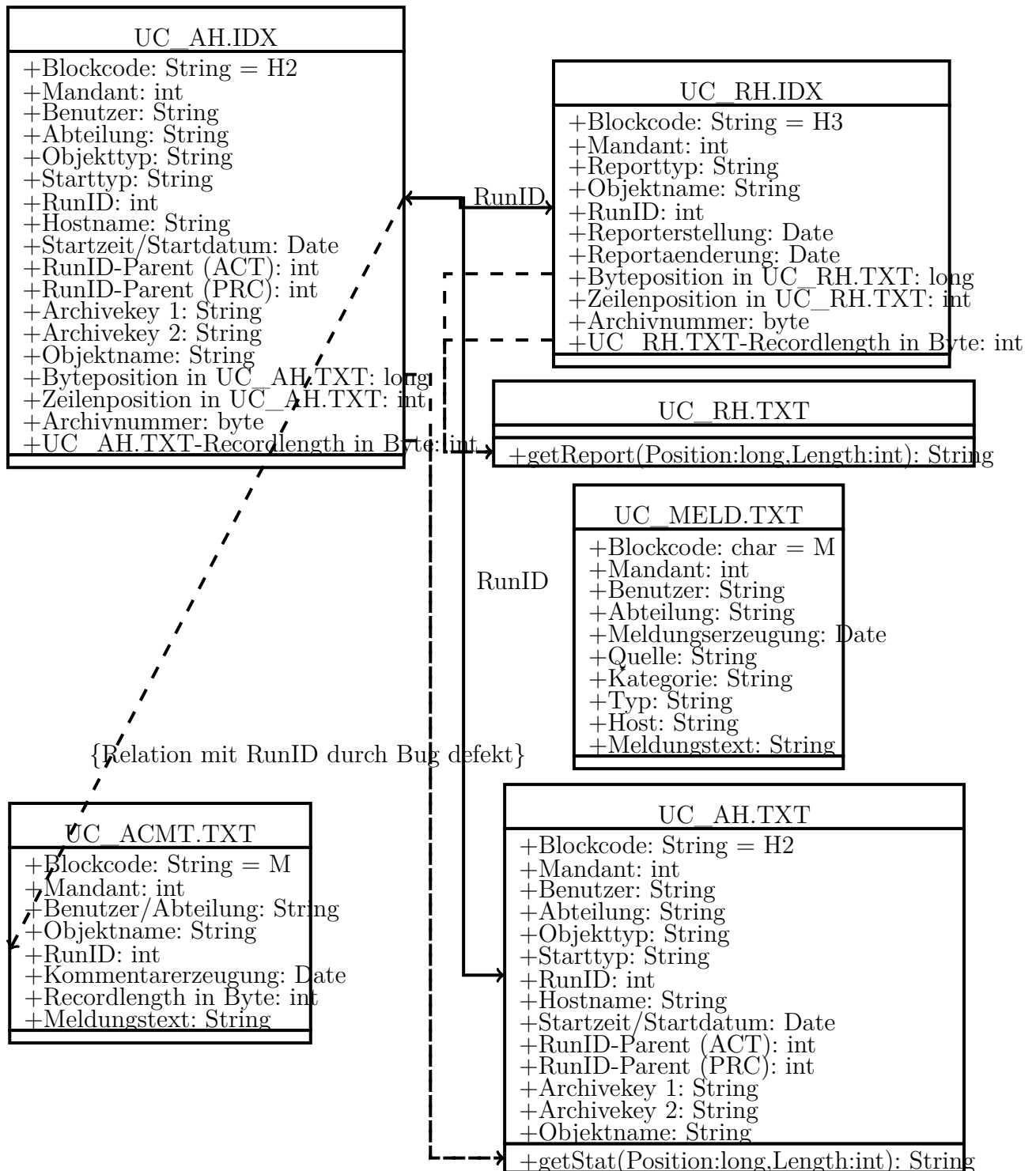


Abbildung 3.1: UML-Diagramm Archivdateien Datenstruktur⁶

⁶Quelle: Jan Thomas Gundlach, 2021 - Dia

Die Datenstruktur der jeweiligen Dateien und deren Abhängigkeiten zueinander sind unter Abb. 3.1 dargestellt.

Listing 3.3: Ermittlung von Zeichenkodierungen archivierter Daten

```
$ file -i ./*
./UC_ACMT.TXT: inode/x-empty; charset=binary
./UC_AH.IDX:   text/plain; charset=us-ascii
./UC_AH.TXT:   text/plain; charset=iso-8859-1
./UC_MELD.TXT: text/plain; charset=iso-8859-1
./UC_RH.IDX:   text/plain; charset=us-ascii
./UC_RH.TXT:   text/plain; charset=iso-8859-1
```

Indexdatei für Ausführungsdaten: UC_AH.IDX

Die Datei UC_AH.IDX dient als Index zum schnellen Zugriff auf die Datei UC_AH.TXT (vgl. 3.1.3). In UC_AH.IDX sind zeilenweise (Lst. 3.4) je Aufgabe die primären Ausführungsdaten durch Strichpunkt getrennt gespeichert sowie die Position in Byte, die Zeilennummer und die Länge der vollständigen Ausführungsdaten aus UC_AH.TXT.

Listing 3.4: Beispielzeile aus Indexdatei für Ausführungsdaten: UC_AH.IDX

```
H2;3000;FISCHER;HANSEMERKUR;SCRI;<PERIOD>;2474844;;2021.05.01 00:31:11;2475290;2475290;;;SC.PERIODICAL_STARTS;
0;1;52;271
```

Zur Gewährleistung der Konsistenz der Daten wurde sichergestellt (Lst. 3.5), dass eine RunID in der Datei UC_AH.IDX nur einmal vorkommt.

Listing 3.5: Überprüfung der Eindeutigkeit einer RunID innerhalb von UC_AH.IDX

```
[NB-2014299] $ awk -F";" '{print $7}' UC_AH.IDX | sort \
| uniq -c | awk '$1>=2{print $0}'
[NB-2014299] $
```

Ausführungsdaten: UC_AH.TXT

In der Datei UC_AH.TXT sind die Ausführungsdaten der archivierten Aufgaben gespeichert. Zum Auslesen der Ausführungsdaten einzelner Aufgaben werden die Positions- und Größenangaben aus UC_AH.IDX (vgl. 3.1.3) benötigt.

Indexdatei für Protokolle: UC_RH.IDX

Die Datei UC_RH.IDX dient als Index zum schnellen Zugriff auf die Datei UC_RH.TXT (vgl. 3.1.3). In UC_RH.IDX sind zeilenweise (Lst. 3.6) je Aufgabe die Protokolleigenschaften durch Strichpunkt getrennt gespeichert sowie die Position in Byte, die Zeilennummer und die Länge der Protokollabschnitte aus UC_RH.TXT.

Je RunID gibt es mehrere, eigenständigen Zeilen.

Listing 3.6: Beispielzeile aus Indexdatei für Protokolle: UC_RH.IDX

```
H3;3000;ACT;SC.PERIODICAL_STARTS;2474844;2021.05.01 00:31:11;2021.05.01 00:31:11;0;1;52;2164
```

Protokolle: UC_RH.TXT

In der Datei UC_RH.TXT sind die Protokolle der archivierten Aufgaben gespeichert. Zum Auslesen der Protokolle einzelner Aufgaben werden die Positions- und Größenangaben aus UC_RH.IDX (vgl. 3.1.3) benötigt.

Je RunID gibt es mehrere, eigenständigen Datensätze hintereinander.

Meldungsarchiv: UC_MELD.TXT

Meldungen enthalten Informationen über den Zustand von ausgeführten Aufgaben oder Prozessstatus. Zur Einordnung werden sie in die drei Prioritätsklassen Fehler, Warnungen und Meldungen eingeteilt.

Die Eigenschaften der Meldungen werden bei der Archivierung durch Strichpunkt getrennt zeilenweise (Lst. 3.7) in die Datei UC_MELD.TXT abgelegt. Meldungen können aufgrund Ihrer archivierten Eigenschaften nicht automatisiert mit dritten Ausführungsdaten in Relation gesetzt werden da ein eindeutiges Kriterium hierfür fehlt.

Listing 3.7: Beispielzeile aus Meldungsarchiv: UC_MELD.TXT

```
M;3000;FISCHER;HANSEMERKUR;2021.04.30 02:00:07;Automation Engine;Nachricht;Information;;  
U00012109 Script 'SC.PERIODICAL_STARTS' (RunID '0002471778') normal beendet.
```

Kommentararchiv: UC_ACMT.TXT

Zu der Ausführung einer Aufgabe kann programmatisch oder manuell ein Kommentar mit individuellen Informationen zur Ausführung hinzugefügt oder abgerufen wer-

den (wie beispielsweise eine Ticket-Referenz bei Aufgabenabbruch). Mit Archivierung werden Kommentare und deren Eigenschaften separat zu den Ausführungsdaten und referenzierbar über die Eigenschaft RunID in die Datei UC_ACMT.TXT abgelegt. Die Ablage von Eigenschaften und Kommentaren erfolgt ebenfalls zeilenweise getrennt durch Strichpunkt (Lst. 3.8).

Innnerhalb der Analyse der Datenstrukturen ist ein Produktfehler im Dienstprogramm AE DB Archive festgestellt worden, welcher die Relation zwischen Ausführungsdaten und Kommentaren unterbindet. Der Softwarefehler wurde beim TIER1-Partner des Herstellers (vgl. [Capture 2021](#)) gemeldet.

Listing 3.8: Beispielzeile aus Kommentararchiv: UC_ACMT.TXT

```
ACMT;3000;FISCHER/HANSEMERKUR;JP.HUM020;140428253184894;2021.05.01 01:25:20;66;  
Der Jobplan wurde durch das Script SC.CLEAN_CANCEL inaktiv gesetzt
```

3.2 Auswahl der Java Implementierung

Theoretisch gibt es mehrere Quellcodes für Java Implementierungen. Praktisch relevant ist inzwischen vermehrt der Quellcode des JDK Project⁷. Auf Basis dieses Quellcodes werden von verschiedenen Anbietern Java Implementierungen angeboten. Diese Anbieter werden umgangssprachlich auch im deutschen Sprachraum häufig als Java Vendor bezeichnet. Die Bezeichnung geht zurück auf interne Systemeinstellungen in Java, in denen der Anbieter einer Java Implementierung über die Eigenschaft `java.vendor` seinen Eigennamen definiert. Die Systemeinstellungen können mittels der Methode `getProperties()` der Klasse `System` ausgelesen werden ([Ziegenbalg 2018](#): Abs. 31).

Java Implementierungen von verschiedenen Anbietern können sich in der Unterstützung verschiedener Prozessorarchitekturen wie x86, ARM, PPC, SPARC und deren unterschiedlichen Wortbreiten unterscheiden (vgl. [Hardware-Zone 2021](#)), als auch in den zu diesen angebotenen Betriebssystemen und deren Distributionen. Darüber hinaus sind aus technischer Sicht die Implementierungen häufig auf bestimmte Anwendungszwecke optimiert und der Ursprungsquellcode entsprechend modifiziert. Neben individuellen Modifizierungen eines Anbieters bzgl. der Implementierung unterscheiden sich die angebotenen Implementierungen auch durch das Beinhalten oder Fehlen von Java Erweiterungen wie beispielsweise Java Mission Control, Java Flight Recorder

⁷Quelle: JDK Project, <http://openjdk.java.net/projects/jdk/>, Aufruf: 30.07.2021

der oder JavaFx bzw. OpenJFX (vgl. [Eclipse Foundation, Inc. 2021](#)).

Abgesehen von technischen Unterschieden wie Unterstützung und direkten Funktionsumfang gibt es auch Unterschiede bzgl. der Wartung. Die Anbieter bieten unterschiedlich frei zugängliche Sicherheitspatches an, veröffentlichen regelmäßige Updates in unterschiedlichen Zyklen und definieren je Version unterschiedlich lange Lebenszyklen. Die Updates selbst werden je nach Anbieter über verschiedene Quellen und Schnittstellen angeboten als auch in verschiedenen Auslieferungsformaten. Insbesondere letztere sind für standardisierte und automatisierte Updateverfahren seitens des Anwenders nicht zu unterschätzen. Es sollte auch betrachtet werden, wie beständig die Weiterentwicklung einer Implementierung ist. Hierzu sollte beispielsweise der Verbreitungsgrad und die Substanz des Anbieters selbst begutachtet werden.

Verbreitung JDK nach Anbietern 2020	
Anbieter	% verwendet
Oracle	74.78
AdoptOpenJDK	7.06
IcedTea	5.30
Azul	2.96
IBM	2.37
Amazon	2.18
Unknown	1.96
Pivotal	1.40
SAP	0.74
Sun	0.58
Debian	0.54
Other	0.10

Tabelle 3.5: Prozentuale Verbreitung von JDK nach Anbietern 2020⁸

Aus organisatorischer und wirtschaftlicher Sicht stellt sich hingegen die Frage, ob ein Anbieter seine Java Implementierung grundsätzlich kostenlos zur Verfügung stellt und ob der Betrieb der JVM in kommerziellen oder öffentlichen, produktiven Umgebungen ebenso unentgeltlich gestattet ist. Insbesondere dann ist aber auch immer von Interesse, ob und zu welchen Konditionen ein Anbieter Unterstützungsleistungen an-

⁸Quelle: New Relic, Inc. - The State of Java: Trends And Data For One of the World's Most Popular Programming Languages 2020, <https://newrelic.com/blog/nerd-life/state-of-java>, Aufruf: 03.08.2021

Verbreitung JDK nach Anbietern 2021		
Anbieter		% verwendet
Oracle Java		59
Generic OpenJDK		36
Adopt OpenJDK		22
Amazon Corretto		10
Azul Zulu		6
GraalVM		4
OpenLogic JDK		4
Other		4

Tabelle 3.6: Prozentuale Verbreitung von JDK nach Anbietern 2021⁹

bietet, ob die Implementierungen zertifiziert sind oder mit welchen Testverfahren die Implementierungen validiert werden (vgl. Azul 2021). Und zuletzt, wie transparent der Implementierungsprozess ist.

Die nachfolgenden Implementierungen wurden aufgrund ihrer Eigenschaften (Evans, Gough, Newland 2018: Abs. 4) und ihres Verbreitungsgrades (Tab. 3.5 und Tab. 3.6) für die Umsetzung als relevant eingestuft und in die Analyse einbezogen.

3.2.1 Amazon Corretto

Corretto (vgl. Amazon Web Services, Inc. 2021) wird seit 2019 von Amazon angeboten und ist damit die jüngste aller Implementierungen. Dabei handelt es sich um eine plattformübergreifende (Linux, macOS, Windows und Docker) Distribution des OpenJDK, welche unter der General Public License (GPL) mit Classpath Exception lizenziert ist und damit kostenlos in kommerziellen Produktionsumgebungen verwendet werden darf. Sie ist zertifiziert nach TCK, wodurch sie offiziell kompatibel zum Java SE Standard ist. Amazon bietet Corretto mit Long Term Support (LTS) inklusive eigenen Leistungsoptimierungen und Sicherheitspatches an. Updates werden mindestens quartalsweise veröffentlicht. Amazon setzt Corretto selbst in vielen produktiven Diensten in der Amazon Web Services (AWS) ein und stellt für den Einsatz innerhalb der AWS Support zur Verfügung.

⁹Quelle: JRebel 2021 Java Developer Productivity Report, <https://www.jrebel.com/system/files/2021-java-developer-productivity-report.pdf>, Aufruf: 10.08.2021

3.2.2 Azul Zulu

Azul bietet unter dem Namen Zulu JDK (vgl. [Azul 2021](#)) eine eigene, kostenlose und plattformübergreifende (Linux, Windows, macOS, Solaris und Docker) OpenJDK Implementierung an. Diese kann ebenfalls kostenlos in kommerziellen Produktionsumgebungen eingesetzt werden. Zur Verfügung stehende Patches zur Korrektur von Sicherheitslücken und Softwarefehlern sind jedoch nur kostenpflichtig über die Azul Platform Core erhältlich (vgl. [Azul 2020](#)). Azul zertifiziert das Zulu JDK mittels TCK und stellt damit offiziell eine Kompatibilität zum Java SE Standard her. Azul hält sich bei der Definition der Versionszyklen an Oracle JDK und verlängert den Lebenszyklus des Zulu JDK jeweils zusätzlich um ein Jahr. Darüber hinaus bietet Azul ein Archiv aller Veröffentlichungen eines Majorrelease innerhalb des Lebenszyklus an. Der kostenlose Zugang zu dem Angebot ist Teil der Zulu Community. Es gibt hierzu jedoch keine explizite Zusicherung seitens Zulu, dieses Angebot langfristig aufrechtzuerhalten (vgl. [Basil Bourque 2021](#)). Azul stellt umfangreiche, kostenpflichtige Supportangebote zur Verfügung. Unter anderem gilt das auch für die Medium Term Support (MTS) Veröffentlichungen sowie eine generelle 24/7-Erreichbarkeit. Das Supportangebot berechnet sich dabei nicht je CPU, sondern je System. Parallel dazu bietet Azul unter Zing eine performanceorientierte Java Plattform an.

3.2.3 Eclipse Temurin

Eclipse Temurin (vgl. [Eclipse Foundation 2021](#)) ist der Nachfolger des AdoptOpenJDK Projektes. Die erste Implementierung wurde am 30. Juli 2021 durch die Adoptium Working Group (vgl. [Eclipse Foundation, Inc. 2021](#)) veröffentlicht. Das Ziel des Temurin Projektes ist die Bereitstellung eines TCK-zertifizierten JDK unter Open-Source-Lizenz zur allgemeinen Verwendung im Java-Ökosystem. Zusätzlich zur TCK-Zertifizierung und der damit verbundenen offiziellen Java SE Kompatibilität durchlaufen die Veröffentlichungen die Eclipse-eigene AQAvit Testinfrastruktur (vgl. [Eclipse Foundation 2021](#)). Veröffentlicht wird Temurin unter der GPL mit Classpath Exception.

Das Temurin Projekt folgt für seine Veröffentlichungen der Roadmap des JDK Project: Neue Versionen mit neuen Features alle 6 Monate, Wartungs- und Sicherheitspatches quartalsweise sowie alle 3 Jahre eine Version mit LTS. Versionen wie mit LTS werden im Gegensatz zu Oracle's OpenJDK jedoch deutlich länger unterstützt und

weiter mit Updates versorgt. Das mindert einen permanenten Migrationsdruck und verringert so den Wartungsaufwand. Hinter der Adoptium Working Group stehen große IT-Konzerne u.a. Alibaba, Azul, Huawei, Karakun, Microsoft, Red Hat und IBM. Da Eclipse als gemeinnützige Gesellschaft nicht die Mittel hat, eigene Sicherheitspatches zu entwickeln, werden diese durch die Mitglieder bereitgestellt.

Für Eclipse Temurin wird durch Eclipse selbst als gemeinnützige Gesellschaft kein kommerzieller Support angeboten. Es gibt lediglich begrenzte Unterstützung bei der Behebung von eindeutig auf die Applikation zurückzuführende Fehler. Azul bietet für Temurin parallel zum hauseigenen Zulu SDK kommerziellen Support an, welcher sich im Umfang vom Support für das Zulu SDK selbst unterscheidet.

3.2.4 IBM OpenJDK

IBM stellt als einziger der näher betrachteten Anbieter eine Virtual Machine auf Basis von J9 (vgl. [IBM 2021](#)) und nicht HotSpot zur Verfügung. Parallel ist es statisch gesehen auch die einzige relevante JVM auf dem Markt, die nicht auf HotSpot aufbaut. J9 ist eine auf geringeren Arbeitsspeicherverbrauch und geringere Start- als auch Anlaufphasen optimierte, plattformübergreifende (AIX, Linux, z/OS und IBM i) JVM. Sie ist damit auch unter anderem insbesondere für den Betrieb in Containern, wie sie beispielsweise in Cloudumgebungen eingesetzt werden, optimiert. IBM hat die J9 Virtual Machine eigens entwickelt und im September 2017 Open-Source als OpenJ9 der Eclipse Foundation zur Verfügung gestellt, wo Implementierungen über das AdoptOpenJDK Projekt bereitgestellt werden. IBM selbst stellt seitdem für J9 nur noch Updates für LTS Veröffentlichungen bereit. Zukünftige LTS Veröffentlichungen sollen nur noch unter OpenJ9 stattfinden. Der Einsatz von OpenJ9 als auch J9 ist in produktiven Umgebungen kostenfrei. Im April 2021 kündigte IBM (vgl. [IBM Developer Blog 2021](#)) an, OpenJ9 TCK-zertifiziert kostenlos über die Adoptium Working Group zu veröffentlichen. Die TCK-Zertifizierung wird überwiegend als Hauptkriterium für die produktive Verwendung einer Java Implementierung im Versicherungsbereich angesehen. Bis zur Abgabe dieser Arbeit war dieses Vorhaben noch nicht umgesetzt. IBM bietet entsprechend kommerziellen Support sowohl für J9 als auch OpenJ9 LTS Veröffentlichung von AdoptOpenJDK an.

Mit Hinblick auf Java 17 als nächste LTS Version im September 2021 soll die Arbeit auf der letzten stabilen Java Version aufbauen, um möglichst geringe Migrationsauf-

wände bei einer Portierung zu Java 17 zu generieren. Das schließt eine Umsetzung auf Grundlage von J9 aus, weshalb das IBM OpenJDK auf Basis von J9 nicht weiter betrachtet wird.

3.2.5 Oracle Java SE

Oracle ist historisch und aufgrund seiner anhaltend hohen Verbreitung der Hauptanbieter von Java Implementierungen. Bei dem Oracle Java SE (vgl. [Oracle Technology Network 2021](#)) handelt es sich in der aktuellen Version um eine plattformübergreifende (Linux, macOS, Windows) JDK Implementierung. Oracles Roadmap sieht vor, alle 6 Monate eine neue Version mit neuen Features zu veröffentlichen, Wartungs- und Sicherheitspatches quartalsweise anzubieten sowie alle 3 Jahre eine Version mit LTS herauszubringen. Das Oracle Java SE bildet den Java SE Standard. Bis 2019 mit der Version 8u202 war die Verwendung in produktiven Umgebungen kostenfrei möglich. Seitdem ist nur noch die Entwicklung mit Oracle Java SE kostenfrei gestattet. Für den Einsatz in produktiven Umgebungen muss eine „Java SE Subscription“ abgeschlossen werden. Damit bietet Oracle parallel auch Support an.

3.2.6 Red Hat OpenJDK

Red Hat bietet mit seinem OpenJDK (vgl. [Red Hat Developer 2021](#)) eine plattformbeschränkte (Linux und Windows) Java Implementierung des Oracle OpenJDK an. Sie ist der Java-Standard des hauseigenen, kommerziellen Betriebssystems Red Hat Enterprise Linux und steht auch in dessen freien Fork CentOS zur Verfügung. Dadurch kann es kostenfrei in produktiven Umgebungen eingesetzt werden. Zusätzlich ist es in verschiedene Produkte der Red Hat Containerwelt integriert. Im Gegensatz zur Konkurrenz verfolgt Red Hat weniger eine breite Palette an unterstützten Plattformen, sondern mehr eine besonders auf die jeweilige Plattform abgestimmte und damit besser integrierte Implementierung (vgl. [Fernando Almeida 2018](#)). Dadurch handelt es sich beim Red Hat OpenJDK um eine modifizierte Implementierung des Oracle OpenJDK.

Updates werden von Red Hat mind. quartalsweise veröffentlicht und analog zur Roadmap des JDK Project als LTS Versionen angeboten. Red Hat bietet für beide Plattformen Support-Optionen an.

Vergleich von JDK Anbietern auf Basis Java > 11					
Eigenschaften	Amazon Corretto	Azul Zulu	Eclipse Temurin	Oracle Java SE	Red Hat OpenJDK
Open Source	X	X	X	-	X
Produktiv kostenlos	X	X	X	-	X
Updates kostenlos	X	X	X	-	X
Updaterythmus	3	3	3	3	3
	Monate	Monate	Monate	Monate	Monate
kommerzieller Support	nur AWS	X	X	X	IBM, Azul
Security-/Bugfixes kostenlos	X	O	X	-	X
Lebenszyklus	6 Jahre	8 + 2 Jahr	6 Jahre	5 + 3 Jahre	6 Jahre
Validierung	TCK	TCK	TCK, Aqavit	TCK	TCK
Betriebssysteme	Linux, Windows, macOS, Docker	Linux, Windows, macOS, Docker	Linux, Windows, macOS	Linux, Windows, macOS	Linux, Windows, OpenShift
Installationsformate	7	6	5	5	4
OpenJFX / JavaFX	-	X	-	X	-
Java Flight Recorder	X	X	-	X	X
Mission Control	-	X	-	X	X
Besonderheit	cloud-ready	JavaFX bundled	Open Source Projekt	Maintainer	Integration

Tabelle 3.7: Vergleich JDK Implementierungen verschiedener Anbietern¹⁰

3.2.7 Ergebnis

Aufgrund der Analyse (Tab. 3.7) wurde Eclipse Temurin gewählt. Eclipse Temurin bietet den Durchschnitt an Unterstützung bzgl. Prozessorarchitektur und Betriebssystemen. Eine Cloudunterstützung wird in der Umsetzung als Client- und Desktopanwendung nicht benötigt. Performancetechnisch sind alle Implementierungen für

¹⁰Quellen: (vgl. Red Hat Developer 2021), (vgl. Amazon Web Services, Inc. 2021), (vgl. Azul 2021), (vgl. Eclipse Foundation 2021), (vgl. IBM 2021)

Azul 2021: OpenJDK vs Oracle JDK - Comparison Table, [https://www.azul.com/products/core/jdk-comparison-matrix/](https://www Azul.com/products/core/jdk-comparison-matrix/), Aufruf: 05.08.2021

die hier betrachtete Anwendung bis auf IBMs J9 gleich. Eine OpenJ9 Variante wurde bis Abgabe der Arbeit jedoch noch nicht TCK-zertifiziert durch die Adoptium Working Group veröffentlicht. Für produktive Umgebungen ist eine TCK-Zertifizierung und damit eine Java SE Kompatibilität oft aufgrund interner Richtlinien und proprietärer Anwendungen maßgeblich. Für J9 als TCK-zertifizierte Variante existiert nur eine Java 11 Implementation. Keine Java 15 oder 16. Da mit Java 17 bereits im September das nächste LTS Release von Java erscheint und der Sprachstandard von Java 16 näher an Java 17 ist als der des letzten LTS Release Java 11, verringert dies in der Regel spätere Migrationsaufwände.

Parallel ist die Verbreitung des AdoptOpenJDK von 2020 auf 2021 in Folge der Lizenzkosten von Oracle Java SE am stärksten gestiegen (Tab. vgl. 3.5 und Tab. vgl. 3.6). AdoptOpenJDK wurde bereits von vielen Fortune 500-Unternehmen (vgl. [Patel, Korando 2019](#)) eingesetzt. Mit der Eclipse Temurin TCK-Zertifizierung wird sich diese Anzahl vermutlich noch erhöhen, da der maßgebliche Nachteil gegenüber anderen Anbietern abgebaut wurde. Neben der TCK-Zertifizierung wird die bereits unter AdoptOpenJDK angewendete Eclipse-eigene AQAvit Testinfrastruktur (vgl. [Eclipse Foundation 2021](#)) weiterhin auch für Eclipse Temurin angewendet.

Eclipse Temurin ist damit nicht nur free for production, sondern auch production-ready, bietet außerhalb des Extended Supports der Konkurrenz im Schnitt gleich lange Lebenszyklen für die veröffentlichten Versionen inkl. Sicherheits-/Wartungspatches und verfügt im Gegensatz zu Azul Zulu über eine langfristige Open Source Strategie. Unter dem Dach der Eclipse Foundation wird dies gesichert durch eine Vielzahl namhafter IT-Konzerne, die hinter dem Projekt stehen. Mittels OpenJFX können Erweiterungen auch zu Eclipse Temurin hinzugefügt werden. Maßgeblicher Vorteil ist, dass das Gesamtpaket frei zur Verfügung steht.

3.3 Auswahl des Java GUI-Framework

Für die Gestaltung der grafischen Oberfläche und Bereitstellung funktional passender Steuerelemente ist für die Umsetzung dieser Arbeit die Auswahl eines GUI-Frameworks notwendig. Im Folgenden werden verschiedene Java GUI-Frameworks dazu betrachtet, um eines als Grundlage für die Arbeit zu definieren.

3.3.1 Swing

Swing (vgl. [Oracle 2021](#)) ist ein UI-Toolkit, welches ursprünglich von Sun entwickelt wurde. Es erweitert das erste native GUI-Framework Abstract Widget Toolkit (AWT) von 1985 um die Swing-Komponenten und bildet zusammen mit dem Java 2D-API, den Unterstützungstechnologien aus Java Accessibility, der Data Transfer-API und AWT die Java Foundation Classes (JFC) zur Erstellung grafischer Benutzeroberflächen (vgl. [Thormählen 2019](#)). Swing-Komponenten bauen zum Teil auf AWT auf und verwenden insbesondere im Event-Handling noch viel AWT-Klassen (vgl. [Petri 2021](#)). Swing zeichnet sich unter anderem durch eine Vielzahl von einfach zu verwendenden Komponenten und Widgets, Unterstützung von Undo-/Redo-Operationen, 2D Grafik Rendering und veränderbarem Look-and-Feel aus.

Swing ist vollständig in Java geschrieben, wodurch es plattformübergreifend nicht von spezifischen Bibliotheken abhängt. Der Nachteil ist, dass das Look-and-Feel einer Swing-Oberfläche sich von dem des nativen Betriebssystems unterscheiden kann. Im Zweifelsfall ist Swing auch von AWT-Komponenten abhängig.

Zeitweise gab es mit Java 9 Überlegungen, Swing zukünftig zum Vorteil von JavaFX aus der Java Standard Edition herauszulösen. Mit Java 11 ist das Gegenteil eingetreten und JavaFX wurde ausgegliedert (vgl. [Smith 2018](#)). Swing ist damit quasi das einzig verbleibende GUI-Framework der Java Standard Edition. Aufgrund der anhaltend hohen Verbreitung ist es teilweise noch Bestandteil der Lehre und wird gerne als Einstieg in die GUI-Entwicklung unter Java verwendet. Es wird jedoch nicht mehr weiterentwickelt oder gewartet.

3.3.2 JavaFX

JavaFX ist der offiziell von Oracle etablierte Nachfolger von Swing (vgl. [Oracle 2018](#)). Noch unter Sun entstand im Jahr 2008 die Version 1. Nach der Übernahme von Sun durch Oracle und einem Redesign von JavaFX mit der Version 2 gewann das Framework zunehmend an Reichweite. Mit Java 8 wurde die Versionierung an die Java Standard Editionsversionen gekoppelt.

Ehemalig sollte mit JavaFx in Verbindung mit Rich Internet Applications und Java Webstart eine Konkurrenz zu Silverlight und Flash aufgebaut werden. Aufgrund der völlig neuen Architektur (Abb. 3.2) ohne Abhängigkeiten zu Swing und AWT bietet es auch abseits davon einen deutlich gesteigerten Funktionsumfang und modernere

Ansätze (vgl. [Kirsch 2021](#)). Dies ermöglicht den Einsatz in verschiedenen Bereichen beispielsweise mit Medieninhalten, mobilen Systemen (iOS, Android) oder in stark von Grafiken geprägten Szenarien.

JavaFX ist als eigenständige Bibliothek entworfen, bringt einen eigenen GUI Builder mit und ermöglicht mittels dem XML-basierten FXML-Format eine Trennung zwischen Darstellung und Logik. Durch die große Anzahl unterschiedlicher und ausgereifter Komponenten, der möglichen Gestaltung über Cascading Style Sheets (CSS) oder der Unterstützung von 3D Darstellungen entspricht es moderneren Ansprüchen ([Ullenboom 2020](#): 1034). Dabei werden mitunter die Fähigkeiten neuerer Grafikprozessoren durch Hardwarebeschleunigung adäquat unterstützt. Der Umstieg von Swing auf JavaFx wird durch die Swing Integration erleichtert und sanfter gestaltet. Zuletzt ermöglicht JavaFX das Erstellen einer Applikation, die keine bestehende Java Installation mehr voraussetzt, indem bei der Paketierung bereits alles Notwendige mit eingebunden wird.

Mit Java 11 entschied sich Oracle, die Weiterentwicklung von JavaFX Open Source in das OpenJFX Projekt (vgl. [OpenJFX Project 2021](#)) unter GPLv2 mit Classpath Exception auszugliedern und aus der Java Standard Edition zu entfernen. Innerhalb des OpenJFX Projektes ist es das Ziel verschiedener Unternehmen und Einzelpersonen, gemeinschaftlich ein vollständiges, modernes und effizientes GUI-Framework bereitzustellen.

3.3.3 SWT

Das Standard Widget Toolkit (SWT) (vgl. [Eclipse Foundation 2021](#)) ist ein von IBM ursprünglich für die Entwicklungsumgebung Eclipse als Ersatz zu AWT entwickeltes GUI-Framework, welches die nativen grafischen Elemente des darunterliegenden Betriebssystems über das Java Native Interface (JNI) verwenden kann. SWT erfordert entsprechend keinerlei Abhängigkeiten zu AWT/Swing, jedoch zu den verwendeten nativen Elementen des Betriebssystems und muss daher Applikationen immer mit systemspezifischen, dynamischen Bibliotheken ausliefern. Die native Anbindung selbst führte anfangs dazu, dass SWT auf dem Zielsystem Windows deutlich performanter als Swing war ([Ullenboom 2020](#): 1036). Plattformübergreifend kann dieser Vorteil jedoch schnell zu einem Nachteil werden, da die Voraussetzungen an die nativen UI-Elemente des Systems zumeist nicht gegeben sind und nachempfunden werden

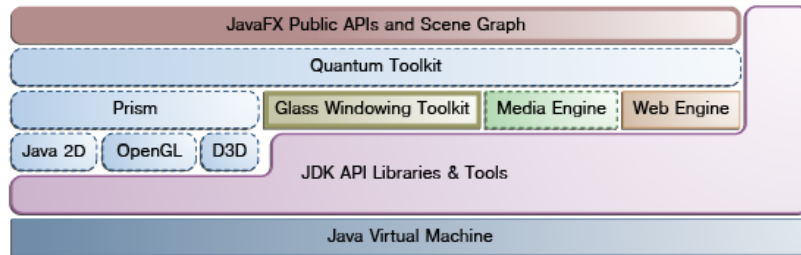


Abbildung 3.2: JavaFX Architektur¹¹

müssen. Parallel kann dies dazu führen, dass es im Zuge der Speicherverwaltung zu einem Mehraufwand in der Entwicklung kommen kann.

SWT verfügt über eine breite Palette an Widgets und bietet die Möglichkeit, optional AWT/Swing Komponenten zu integrieren (vgl. [Eclipse Contributors 2021](#)). Durch die Verwendung nativer UI-Elemente des Betriebssystems kann sich die Oberfläche und die Handhabung einer SWT-Applikation mehr an die Optik und das Bedienerlebnis des Betriebssystems anpassen. Die Entwicklung von SWT wird seit deren Gründung durch die Eclipse Foundation fortgeführt. SWT ist von dieser als Open Source unter der Eclipse Public License (EPL) 2.0 lizenziert und gestattet damit in der Regel eine kommerzielle Nutzung.

3.3.4 Ergebnis

Unter Swing ist die Umsetzung der Arbeit deutlich aufwendiger als unter JavaFX oder SWT, da eine TreeTable-Komponente kein elementarer Bestandteil von Swing ist. Eine solche Komponente müsste in Swing mit entsprechendem Aufwand eigens erstellt, erweitert und gewartet werden. Mit der Einstellung der Weiterentwicklung von Swing und der Definition von JavaFX als Nachfolger durch Oracle sollten Neuentwicklungen in der Regel nicht mehr unter Swing durchgeführt werden. Aus diesen Gründen wird Swing für die Auswahl eines GUI-Frameworks nicht weiterverfolgt. Eine durch JavaFX oder SWT übersetzte Applikation kann in beiden Fällen nicht plattformunabhängig verwendet werden. In verschiedenen Veröffentlichungen zu dem Thema wird davon ausgegangen, dass sich JavaFX durchsetzen wird ([Schildt 2014](#): 1105), ([Ullenboom 2020](#): 1036). Hintergrund sind der geringere Entwicklungsaufwand, die

¹¹Quelle: Oracle 2014 - JavaFX: Getting Started with JavaFX, <https://docs.oracle.com/javase/8/javafx/get-started-tutorial/jfx-architecture.htm#JFXST788>, Aufruf: 16.07.2021

neuere Architektur sowie der damit einhergehende Funktionsumfang. Insbesondere gilt dies für mobile oder eingebettete Systeme.

Vergleicht man die Aktivität der Repositorien, weist SWT (vgl. [Eclipse Foundation 2021](#)) mehr als doppelt so viele wie JavaFX (vgl. [GitHub 2021](#)) auf. Dabei ist zu beachten, dass etwa die Hälfte der Commits des SWT Repositoriums dem Eclipse Releng Bot entstammen und damit keine direkte Aktivität für das Repository darstellen. Eine aktive Weiterentwicklung ist für beide Frameworks gegeben.

Da im Codevergleich in der Regel weniger Zeilen für eine JavaFX-Anwendung als für eine SWT-Anwendung benötigt werden, weist SWT eine abstraktere Logik auf, SWT durch die native Anbindung zu einer komplexeren Entwicklung sowie zu höherem Portieraufwand führen kann und JavaFX das modernere Framework ist, fällt die Wahl auf JavaFX als GUI-Framework.

3.4 Java Dateieingangsverarbeitung

In Java gibt es innerhalb der Pakete `java.io` und `java.nio.file` unterschiedliche Klassen, deren Funktionalität sich je nach gewünschter Dateioperation mehr oder weniger eignet. Um das Spektrum geeigneter Klassen für die jeweiligen Dateioperationen einzuschränken, werden anhand der Fragestellungen aus Tab. 3.8 die einzusetzenden Klassen ermittelt.

3.4.1 Klasse/Methode für Protokolle

- Wenn Daten nur gelesen werden müssen, sind lediglich Reader-Klassen/-oder Methoden notwendig.
- Für rein textbasierte Daten müssen Binäroperationen nicht zwangsweise herangezogen werden.
- Kleine, vordefinierte Datenbestände ab einer bestimmten Position sollten innerhalb einer Anweisung in einem Stück gelesen werden, um Zwischenschritte in der Verarbeitung zu vermeiden und die Zugriffe auf das Dateisystem möglichst konstant zu halten.

¹²Quellen: (vgl. [Woltmann 2019](#)), (vgl. [Woltmann 2020](#))

Definition Dateioperation und zugehörige Klassen/Methoden		
Fragen	Ausführungsdaten	Protokolle
Ist der Zugriff lesend, schreiben oder beides?	lesend	lesend
Handelt es sich um Binär- oder Textdaten?	Text	Text
Sind die Daten klein oder groß?	groß	klein
Sollen die Daten zeilen-, zeichen- oder byteweise verarbeitet werden?	zeilenweise	byteweise
Soll von Anfang oder ab einer bestimmten Position begonnen werden?	Anfang	Position
Sollen Daten in einer Schleife oder Anweisungen verarbeitet werden?	Schleife	Anweisung
Klassen/Methoden ¹²	<input type="text" value="InputStreamReader"/> <input type="text" value="BufferedReader"/> <input type="text" value="BufferedReader.readLine"/> <input type="text" value="Files.newBufferedReader"/>	<input type="text" value="FileChannel"/>

Tabelle 3.8: Fragestellungen bzgl. Dateioperationen zur Ermittlung einsetzbarer Klassen oder Methoden

Hierdurch lassen sich die relevanten Klassen für die Protokolle auf die unter Tab. 3.8 Klassen/Methoden eingrenzen. Die Analyse der Dateioperationen für die Protokoll-daten hat die Klasse `FileChannel` zum Lesen der Protokolle aus den Archivdateien ergeben.

3.4.2 Klasse/Methode für Ausführungsdaten

- Wenn Daten nur gelesen werden müssen, sind lediglich Reader-Klassen/-oder Methoden notwendig.
- Für rein textbasierte Daten müssen keine Binäroperationen herangezogen werden.
- Große Datenbestände sollten gepuffert (buffered) verarbeitet werden, sofern die Verarbeitungseinheit nicht der Blockgröße selbst entspricht oder ein Vielfaches

davon ist. Bei gepufferten Dateioperationen werden die Daten des Dateisystems in der unterliegenden Schicht blockweise verarbeitet und in einen Zwischenspeicher (Puffer) geladen. Das verringert die effektiven Zugriffe auf das Dateisystem und erhöht somit die Geschwindigkeit, da die anschließende Verarbeitung aus dem Arbeitsspeicher deutlich schneller ist als aus dem Dateisystem. Optimalerweise sollte die Blockgröße zu der des Dateisystems passen. Durch eine Doppel- oder Dreifachpufferung kann der Effekt noch erhöht werden. Die intern verwendete Standard-Blockgröße von den aufgeführten Klassen beträgt in Java 8 Kilobyte.

- Zeilenweises Auslesen der Daten verringert die I/O-Zugriffe und erhöht damit die Performanz.
- Wenn Daten aus einer Quelle von Anfang an gelesen werden sollen, wird von der eingesetzten Klasse keine Positionierungsmethode verlangt.
- Wenn Daten in einer Schleife verarbeitet werden, sollten insbesondere große Daten nicht ungefiltert in Datenstrukturobjekte überführt werden, um den Arbeitsspeicher zu schonen.

Die Verarbeitung der Ausführungsdaten erfolgt mittels `BufferedReader.readLine` am schnellsten (vgl. Abb. 3.3), weshalb das Lesen der Ausführungsdaten aus den Archivdateien durch `BufferedReader.readLine` umgesetzt wird.

¹³Quelle: Sven Woltmann - Dateien in Java, Teil 1: Übersicht Performance, https://www.happycoders.eu/de/java/dateien-einfach-schnell-lesen/#Ubersicht_Performance, Aufruf: 16.07.2021

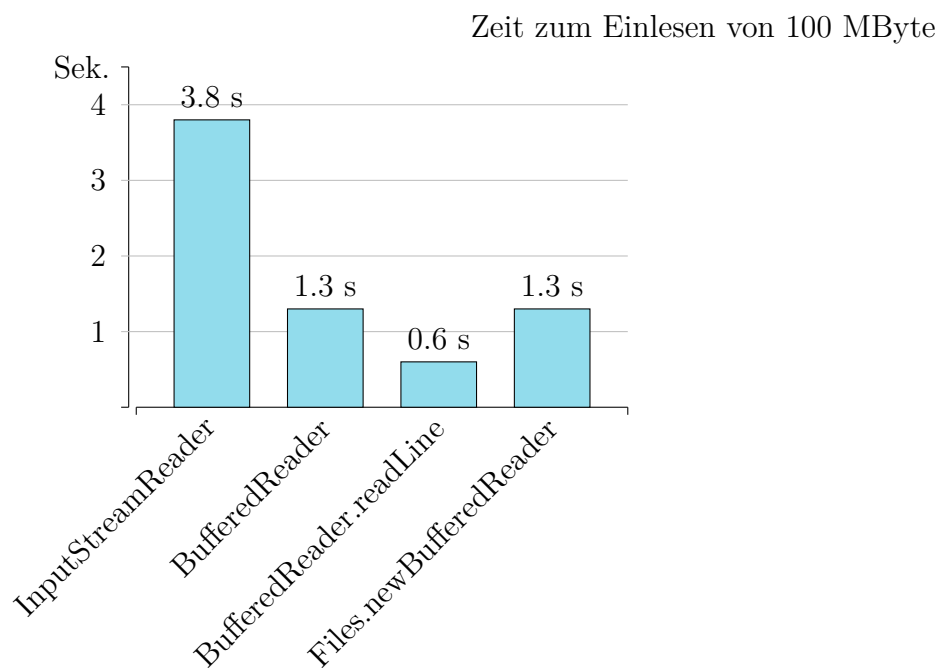


Abbildung 3.3: Vergleich der Zeiten zum Einlesen einer 100 Millionen Byte großen Datei in Java (buffered)¹³

4 Implementierung

4.1 Paketierung

Java ist eine verbreitete Plattform, auf der auch andere Desktopanwendungen aufbauen. Dadurch ist meist eine JVM bereits auf dem System eines Nutzers vorhanden. Dabei ist nicht ungewöhnlich, dass die Version auf dem System des Nutzers nicht zu der Anforderung einer neuen Anwendung passt. Oft kann die JVM aber nicht einfach aktualisiert werden, da bestehende Programme genau diese Version verlangen und für höhere Versionen z.B. noch nicht freigegeben sind. Daher wurde eine Lösung favorisiert mittels welcher das relationsorientierte Recherchewerkzeug unabhängig von dieser Problematik existieren kann. Mit JLink kann dies erreicht werden. JLink baut bei der Paketierung eine Gruppe von Modulen zusammen und erzeugt ein Java-Laufzeitumgebung (JRE) Image. Dies beinhaltet eine client-spezifische JRE, welche in die Anwendung eingebettet ist. Oracle empfiehlt diese Strategie seit für Desktop-Anwendungen seit 2018 ([Oracle Corporation 2018: 6](#)).

Das Aufspielen, Aktualisieren oder Entfernen des relationsorientierten Recherche-werkzeugs kann so unabhängig zu anderen Java-Applikationen in der Anwenderland-schaft durchgeführt werden. Einziger Nachteil ist der erhöhte Festplattenbedarf durch die eigene JRE von rund 100 Megabyte. Diese wird in Zeiten von Giga- und Terabyte-Speichergrößen nicht als problematisch angesehen.

4.2 Klasse ArchiveData

Diese Klasse (Lst. 4.1) bietet Methoden zum Laden von Atomic Automation Engine Archivdaten und Überführen dieser in verknüpfte Adjazenzmengen und eine Streu-werttabelle. Die Adjazenzmengen dienen zusammen mit der Streuwerttabelle Dritt-klassen zur Darstellung der Archivdaten in einer Baumstruktur. Zusätzlich steht eine

4 Implementierung

Methode zum Laden von Protokollen aus den Archivdaten zur Verfügung, die von den jeweiligen Knoten aus der Streuwerttabelle referenziert werden.

Listing 4.1: ArchiveData Klassenkopf mit Attributen

```
public class ArchiveData {

    // Anhand Hashmap verknuepfte Adjazenzmengen
    private HashMap<Integer,HashSet<Integer>> parents = new HashMap<>();
    // Streuwerttabelle Knoten mit Referenz auf Ausfuehungsdaten
    private HashMap<Integer,String[]> runids = new HashMap<>();

    // Indexkonstanten zum verwalten und zugreifen der
    // Ausfuehungsdaten der run-ID Knoten der Streuwerttabelle
    public final int AH_CLIENT = 0;
    public final int AH_OBJECTTYPE = 1;
    public final int AH_RUNID = 2;
    public final int AH_HOSTNAME = 3;
    public final int AH_STARTDATE = 4;
    public final int AH_ARCHIVKEY1 = 5;
    public final int AH_ARCHIVKEY2 = 6;
    public final int AH_OBJECTNAME = 7;
    public final int AH_REPORTPOS = 8;
    public final int AH_REPORTSIZE = 9;
    public final int AH_ARCHIVPATH = 10;

    // Automic Automation Engine Archivdateinamen
    private final String AH_INDEX_FILE = "UC_AH.IDX";
    private final String RH_REPORT_FILE = "UC_RH.TXT";
    private final String RH_INDEX_FILE = "UC_RH.IDX";
    ...
}
```

Konstruktor ArchiveData

Der Konstruktor (Lst. 4.2) initialisiert dieses Objekt nach dem instanzieren. Er stösst das rekursive Suchen nach Indexdateien und das Einlesen dieser in die Datenstruktur dieser Klasse durch Folgeaufrufe an.

Listing 4.2: Konstruktor der Klasse ArchiveData

```
public ArchiveData(File rootPath) {
    // Auslesen des ausgewählten Ordners
    walkArchiveTree(rootPath);
}
```

4.2.1 Methoden

getParents

Diese Methode (Lst. 4.3) stellt die verknüpften Adjazenzmengen Drittklassen zur Verfügung.

Listing 4.3: getParents Methodenkopf

```
public HashMap<Integer,HashSet<Integer>> getParents();
```

getRunids

Diese Methode (Lst. 4.4) ermöglicht Drittklassen einen fortwährenden Zugriff auf alle RunID-Knoten und deren Eigenschaften mittels einer Streuwerttabelle.

Listing 4.4: getRunids Methodenkopf

```
public HashMap<Integer,String[]> getRunids();
```

loadReport

Diese Methode (Lst. 4.5) ermöglicht das Laden eines archivierten Aufgabenprotokolls anhand einer RunID. Sie greift dazu mittels der RunID über die Streuwerttabelle auf die Positionsdaten der Aufgabe zu und liest das angefragte Protokoll aus dem Protokollarchiv.

Listing 4.5: loadReport Methodenkopf

```
public String loadReport(int runId);
```

walkArchiveTree

Diese Methode (Lst. 4.6) öffnet einen ausgewählten Ordner im Dateisystem und durchsucht ihn rekursiv nach Indexdateien nebensiegend Ausführungsdaten- oder Protokollarchiven. Gefundene Indexdateien werden über Drittmethode ausgelesen sowie der Streuwerttabelle und den Adjazenzmengen über diese hinzugefügt. Diese Methode ist nicht sicher gegen nicht alphabetisch sortierten Ordnerstrukturen.

Listing 4.6: walkArchiveTree Methodenkopf

```
private void walkArchiveTree(File rootPath);
```

collectDataStatics

Diese Methode (Lst. 4.7) lädt eine gefundene Ausführungsdaten-Indexdatei und ermittelt alle Parent-Aufgaben sowie deren Adjazenzmengen. Parallel liest sie je Ausführungsdaten-Indexdatei die zugehörigen Protokollarchiven.

rungeintrag (Zeile) alle als Indexkonstanten definierten Spalten unter dem Eintrag der Aufgabe in der Streuwerttabelle ein.

Listing 4.7: collectDataStatics Methodenkopf

```
private void collectDataStatics(Path path);
```

collectDataReports

Diese Methode lädt eine gefundene Protokoll-Indexdatei und ermittelt alle Positionsdaten zu den Protokollen der Aufgaben. Die gefundenen Positionsdaten werden den jeweiligen Aufgabeneinträgen in der Streuwerttabelle hinzugefügt. (Lst. 4.8)

Listing 4.8: collectDataReports Methodenkopf

```
private void collectDataReports(Path path);
```

4.3 Klasse ArchiveTreeTable ¹

Diese Klasse (Lst. 4.9) erstellt ein Fenster mit Steuerelementen zur Darstellung von Automic[®] Automation Engine Archivdaten in einer tabellarischen Baumstruktur inklusive des jeweiligen Protokolls in Textform. Darüber hinaus bindet die Klasse Logiken an die Steuerelemente. Darüber wird das kombinierte sowie spaltenübergreifende Suchen in der Baumstruktur mittels Leerzeichen getrennter Schlagwörter sowie das Markieren der markierten Spalte währenddessen ermöglichen. Zum Laden der Automic Automation Engine Archivdaten greift diese Klasse auf die Klasse ArchiveData zu.

Listing 4.9: ArchiveTreeTable Klassenkopf

```
public class ArchiveTreeTable extends Application {  
  
    private TreeItem<Map<String, Object>> treeRoot;  
    private TreeTableView<Map<String, Object>> treeView;  
    private TreeItem<Map<String, Object>> currentChild;  
    private TreeItem<Map<String, Object>> currentRootChild;  
    private ArchiveData archiveData;  
    ...  
}
```

¹Quelle: Abgeleitet (vgl. [rli 2014](#))

4.3.1 start

Diese Methode (Lst. 4.10) muss durch die geerbte Klasse **Application** implementiert werden. Sie dient zur Initialisierung der Klasse und zeichnen der Steuerelemente und wird in Folge des Methodenaufrufes `launch(args)` aufgerufen.

Listing 4.10: start Methodenkopf

```
@Override  
public void start(Stage primaryStage) throws Exception;
```

expandBranch

Diese Methode (Lst. 4.11) klappt rekursiv im Rücklauf alle Elternzweige eines übergebenden Zweiges aus.

Listing 4.11: expandBranch Methodenkopf

```
private void expandBranch(TreeItem<Map<String, Object>> branch);
```

searchChild

Diese Methode (Lst. 4.12) sucht rekursiv nach dem Nachfolgeelement eines Elements nach Neuinitialisierung des Baumes, da die Elemente dann auf Objektebene nicht mehr gleich sind. Als eindeutiges Vergleichskriterium wird die RunID verwendet. Eine Neuinitialisierung des Baumes tritt durchs Suchen auf. Die Kenntnis über das Nachfolgeelement ist zum Merken des markierten Elements über das Suchen hinweg notwendig.

Listing 4.12: searchChild Methodenkopf

```
private TreeItem<Map<String, Object>> searchChild(TreeItem<Map<String, Object>> root, TreeItem<Map<String,  
Object>> searchedChild);
```

searchAgain

Diese Methode (Lst. 4.13) löst einen Suchvorgang durch Veränderung der Eingabe im Suchtextfeld aus. Zusätzlich sorgt sie dafür, dass die Sortiereinstellung und Zeilenmarkierung nach Abschluss des Suchvorgangs erhalten bleibt.

Listing 4.13: searchAgain Methodenkopf

```
private void searchAgain(String searchString);
```

searchInTree

Diese Methode (Lst. 4.14) durchläuft den Baum und erstellt anhand des Suchbegriffs einen reduzierten Baum. Es werden nur Aufgaben oder deren Oberaufgaben hinzugefügt in denen alle Schlagwörter in beliebigen Eigenschaften vorkommen. Parallel wird zu der zuletzt markierten Aufgabe gescrollt und sie erneut markiert, sofern sie sich in der Suchmenge befindet.

Listing 4.14: searchInTree Methodenkopf

```
private TreeItem<Map<String, Object>> searchInTree(TreeItem<Map<String, Object>> root, String searchString, TreeItem<Map<String, Object>> searchTree);
```

isContains

Diese Methode (Lst. 4.15) prüft ob jedes Schlagwort mind. einmal in irgendeiner Aufgabeneigenschaft vorkommt. Die Methode stellt damit den Abgleich für die Funktionalität zum kombinierten und übergreifenden Suchen da.

Listing 4.15: isContains Methodenkopf

```
private boolean isContains(Map<String, Object> properties, String[] searchArray);
```

buildTaskTree

Diese Methode erstellt rekursiv einen neuen Aufgabenbaum anhand von Daten aus einem Ausführungsdaten- und Protokollarchiv. (Lst. 4.16)

Listing 4.16: buildTaskTree Methodenkopf

```
private void buildTaskTree(Integer runId, TreeItem<Map<String, Object>> newTaskTree);
```

makeTask

Diese Methode (Lst. 4.17) erstellt eine neue Aufgabe, hängt diese in den übergebenden Aufgabenzweig und gibt die neue Aufgabe an den Aufrufer als Ergebnis zurück.

Listing 4.17: makeTask Methodenkopf

```
private TreeItem<Map<String, Object>> makeTask(TreeItem<Map<String, Object>> taskBranch, String
col_objectname, String col_startdate, long col_runid, int col_client,
String col_objecttype, String col_hostname, String col_archivkey1, String col_archivkey2);
```

newColumn

Diese Methode (Lst. 4.18) erstellt eine neue Spalte für eine Aufgabeneigenschaft im Aufgabenbaum.

Listing 4.18: newColumn Methodenkopf

```
protected void newColumn(String name, String colIndex);
```

main

Statische Methode (Lst. 4.19) für die JVM über welche diese die Klasse instanziiieren kann.

Listing 4.19: main Methodenkopf

```
public static void main(String[] args);
```

4.4 Datenstruktur

4.4.1 Aufbau der Baumstruktur

In der vorgestellten Archivdatenstruktur (vgl. 3.1.3) der Automic® Automation Engine kann über die RunID und die Parent RunID (ACT) aus der Indexdatei für Ausführungsdaten (vgl. 3.1.3) eine Baumstruktur erstellt werden mit einer logischen Ursprungsaufgabe als Wurzel. Hierzu werden alle Parent Runids in einer Streuwerttabelle mit ihren Adjazenzmengen verknüpft. In einer weiteren Streuwerttabelle werden alle RunIDs mit Ihren Ausführungsdaten in Form eines Feldes verknüpft. Zuletzt wird die Differenzmenge von Parent-RunIDs und RunIDs bestimmt und darüber alle Parent-RunIDs bestimmt deren Parent-RunIDs selbst nicht bekannt sind. Diese verbleibenden Parent-RunIDs bilden zusammen mit den bereits ermittelten RunIDs der Tiefe 1 das Niveau 1 des Baumes. Von dieser Ebene ausgehend können anhand der Streuwerttabelle mit den Parent RunIDs nun alle nachfolgenden Ebenen ermittelt werden.

4.4.2 Darstellung der Baumstruktur

Die Baumstruktur kann in JavaFx durch das Control `javafx.scene.control.TreeTableView` dargestellt werden. Der Baum besteht dabei aus Objekten von `javafx.scene.control.TreeItem`. Jedes `TreeItem` bildet einen Knoten. Von der Wurzel mit der logischen RunID 0 aus werden dabei alle RunID-Kinds-knoten einer Adjazenzmenge eines Parent-RunID `TreeItem` Objekts diesem via `TreeItem.getChildren().add(child)` hinzugefügt. Zum Schluss wird das Wurzel `TreeItem` Objekt im `TreeTableView` Control als Wurzel definiert und so der Baum im `TreeTableView` dargestellt.

4.5 Klassendiagramm

4.6 Ergebnis

Der Prototyp des relationsorientierten Recherchewerkzeugs für Automic® Automation Engine Archivdaten verfügt über diese Merkmale:

- 1. Mittels der Schaltfläche öffnen kann ein Ordner ausgewählt werden unterhalb von dem rekursiv nach beliebigen Indexdateien für Ausführungsdaten- oder Protokolle gesucht wird. Jedes gefundene Archiv wird zur Darstellung geladen.
- 2. Die geladenen Aufgaben und Ihre Ausführungen werden in Relation mit Ihren Ober- und Unteraufgaben in einer Baumstruktur dargestellt.
- 3. Im Standard sind der Übersicht halber alle Aufgaben eingeklappt. Die jeweiligen Unteraufgaben können durch Anklicken der Pfeilspitzen aufgeklappt werden.
- 4. Die gesamte Liste kann mittels rechtsbündiger Scrollbar durchgescrollt werden.
- 5. Um die angezeigten Aufgaben und deren Relationen einzugrenzen können die Aufgaben durch Eingaben von Schlagworten in das Textfeld oberhalb der Baumstruktur durchsucht werden. Durch Leerzeichen getrennte Schlagworte werden als eigenständige Suchbegriffe erkannt. Das Suchkriterium für eine Aufgabe gilt als erfüllt, wenn jedes Schlagwort innerhalb einer beliebigen Spalte

einer Aufgabe gefunden wird. Die Suche ist unabhängig von Groß- und Kleinschreibung.

- **6.** Das Markieren einer Aufgaben wird von dem Prototyp bis zum Wechsel der Markierung gespeichert, so dass eine Markierung auch nach der Eingrenzung mittels Schlagwort oder des Entfernens eines Suchbegriffs weiter angezeigt wird. Die Scrollbar folgt bei der Suche der markierten Eintragung.
- **7.** Zugeklappte Unteraufgaben werden, sofern Sie Teil der Treffermenge einer Suche sind, bis zur Unteraufgabe selbst automatisch erweitert.
- **8.** Mit dem Selektieren einer Aufgabe wird das jeweilige Protokolle der Aufgabe, sofern verfügbar, in das darunterliegende Textfeld geladen.
- **9.** Anhand der rechtsbündigen Scrollbar des Textfeldes können mehrzeilige Protokolle durchgescrollt werden.
- **10.** Innerhalb ein und derselben Sitzung können durch das Dienstprogramm immer wieder andere Archivdaten geladen werden ohne dass ein Neustart der Applikation notwendig ist.

²Quelle: Jan Thomas Gundlach, 2021 - Screenshot

4 Implementierung

Objectname	Startdate	RunID	Client	Objecttype	Hostname	Archivkey1	Archivkey2
JP.RZS98N	2021.08.12 06:00:03	2871787	8000	JOBP			ADM
JP.RZS98D	2021.08.12 06:00:03	2870860	8000	JOBP			ADM
JP.RZS98S	2021.08.12 06:00:03	2871788	8000	JOBP			ADM
JP.RZS90F	2021.08.12 12:00:04	2872337	8000	JOBP			ADM
JP.RZS99DF	2021.08.12 12:00:04	2873336	8000	JOBP			ADM
JP.RZS99SF	2021.08.12 12:00:05	2873343	8000	JOBP			ADM
P.JSCH.ADM.21_06	2021.08.12 00:00:00	2836357	3000	PARENT			
JP.SCH.ADM.21_06	2021.08.12 00:00:02	2870558	3000	JSCH			
JP.RZS98D	2021.08.12 06:00:03	2870863	3000	JOBP			ADM
JP.RZS98S	2021.08.12 06:00:03	2870862	3000	JOBP			ADM
JP.RZS98N	2021.08.12 06:00:03	2871790	3000	JOBP			ADM
JP.RZS90F	2021.08.12 12:00:04	2873329	3000	JOBP			ADM
JP.RZS99SF	2021.08.12 12:00:05	2872343	3000	JOBP			ADM
JP.RZS99DF	2021.08.12 12:00:04	2873335	3000	JOBP			ADM

Log of operations:

- 2021.08.12 12:00:04:U00020237 Die Objektvariable '&P_C_OWNER#' wurde mittels 'PSET' im Objekt 'JADMIN.KONVERT.SYSTEMPARAMETER' Zeile '00055' (RunID: '0002873336') mit dem Wert '' erstellt.
- 2021.08.12 12:00:04:U00020206 Variable '&P_C_OWNER#' wurde mit Wert '' gespeichert.
- 2021.08.12 12:00:05:U00020206 Variable '&P_C_NETWORK#' wurde mit Wert '' gespeichert.
- 2021.08.12 12:00:05:U00020237 Die Objektvariable '&P_C_JOB#' wurde mittels 'PSET' im Objekt 'JADMIN.KONVERT.SYSTEMPARAMETER' Zeile '00057' (RunID: '0002873336') mit dem Wert '' erstellt.
- 2021.08.12 12:00:05:U00020206 Variable '&P_C_JOB#' wurde mit Wert '' gespeichert.
- 2021.08.12 12:00:05:U00020237 Die Objektvariable '&P_C_RUN#' wurde mittels 'PSET' im Objekt 'JADMIN.KONVERT.SYSTEMPARAMETER' Zeile '00058' (RunID: '0002873336') mit dem Wert '0' erstellt.
- 2021.08.12 12:00:05:U00020206 Variable '&P_C_RUN#' wurde mit Wert '0' gespeichert.
- 2021.08.12 12:00:05:U00020237 Die Objektvariable '&P_C_SUFFIX#' wurde mittels 'PSET' im Objekt 'JADMIN.KONVERT.SYSTEMPARAMETER' Zeile '00059' (RunID: '0002873336') mit dem Wert '' erstellt.
- 2021.08.12 12:00:05:U00020206 Variable '&P_C_SUFFIX#' wurde mit Wert '' gespeichert.

Abbildung 4.1: Prototyp eines relationsorientierten Recherchewerkzeugs für Automic® Automation Engine Archivdaten²

5 Fazit und Ausblick

5.1 Fazit

rechtliche rahmenbedingungen waren ausschweifend Analyse der Software hat unbekannten Bug zum Vorschein gebracht Die ursprüngliche Vorstellung nur einen Baum verknüpfter RunIDs im Arbeitsspeicher zu halten und alle anderen Ausführungsdaten ausschließlich bei Bedarf nachzuladen konnte aufgrund der Schnittstellen von `javafx.scene.control.TreeItem` und `javafx.scene.control.TreeTableView` in einem Prototypen nicht umgesetzt werden. Ein erhöhter Arbeitsspeicherbedarf wurde daher vorerst in Kauf genommen. In Tests ergab sich dieser jedoch insbesondere beim durchsuchen des Aufgabenbaumes oder beim neu laden von Archivdaten als überproportional hoch. Daher wurden für den Testbetrieb übergangsweise Garbage Collection Anreize an diesen Stellen in den Quellcode implementiert. Bei großen und komplexen Archivdaten kam es seitens des `TreeTableView` zu reproduzierbaren Fehlanzeigen einiger spezieller `TreeItems` in Verbindung mit zwischenspeichern der Sortiervorgaben nach einem Suchvorgang. Die Analyse diesbzgl. ist bisher ergebnislos. Ggf. hängt der Fehler mit einem nicht korrekt parametrisierten, generischem Objekt zusammen.

5.2 Ausblick

5.2.1 Bugs

Eine genaue Analyse zum Hintergrund steht noch aus.

5.2.2 Features

ansicht mit kommentaren sobald fehler behoben hinzufügen von reportgröße als spalte

5.2.3 DORA

Digital Operational Resilience Framework for financial services (DORA) sieht vor eine Meldepflicht für den gesamten Finanzsektor zu etablieren. Das DORA ist ein Legislativvorschlag der EU-Kommission, der allerdings erst am 24. September 2020 veröffentlicht wurde und derzeit von den Mitgliedstaaten verhandelt wird.

Mit einer Beschlussfassung ist noch im Jahr 2021 zu rechnen, das Gesetz wird dann voraussichtlich 2022 in Kraft treten.

Abbildungsverzeichnis

2.1	Prozessautomatisierung mit einer Process Engine	14
2.2	Job-Beispiel anhand eines generischen Jobs	16
2.3	Beispiel eines Workflows am Reorganisationsprozess	17
2.4	Eine Liste von Ausführungen und deren Abhängigkeiten	18
2.5	Ansicht archivierter Datensätze im Automic® Archive Browser	19
3.1	UML-Diagramm Archivdateien Datenstruktur	40
3.2	JavaFX Architektur	53
3.3	Vergleich der Zeiten zum Einlesen einer 100 Millionen Byte großen Datei in Java (buffered)	57
4.1	Prototyp eines relationsorientierten Recherchewerkzeugs für Automic® Automation Engine Archivdaten	67

Abkürzungsverzeichnis

VB Visual Basic

VBA Visual Basic for Applications

BPM Business Process Management

EABPM European Association of Business Process Management

CBOK® Common Body of Knowledge

HLL Höheren Programmiersprache (High-level language)

AWT Abstract Widget Toolkit

JFC Java Foundation Classes

WP Arbeitsprozesse

PWP Primären Arbeitsprozess

CP Kommunikationsprozesse

DWP Dialogprozesse

JWP Java-Arbeitsprozesse

JCP Java-Kommunikationsprozess

CSS Cascading Style Sheets

SWT Standard Widget Toolkit

JNI Java Native Interface

EPL Eclipse Public License

ITSiG IT-Sicherheitsgesetz

BSI-KritisV Verordnung zur Bestimmung Kritischer Infrastrukturen nach dem BSI-Gesetz

BSiG Gesetz über das Bundesamt für Sicherheit in der Informationstechnik

BSI Bundesamt für Sicherheit in der Informationstechnik

KRITIS Kritische Infrastruktur

DSGVO Datenschutz-Grundverordnung

VAIT Versicherungsaufsichtlichen Anforderungen an die IT

VAG Versicherungsaufsichtsgesetz

DORA Digital Operational Resilience Framework for financial services

BaFin Bundesanstalt für Finanzdienstleistungsaufsicht

JLS Java Language Specification

JCP Java Community Process

JSR Java Specification Request

TCK Technology Compatibility Kit

JDK Java Development Kit

JVM Java Virtual Machine

JRE Java-Laufzeitumgebung

LTS Long Term Support

MTS Medium Term Support

AWS Amazon Web Services

GPL General Public License

AWI Automic[®] Web Interface

GUI Graphical User Interface

Tabellenverzeichnis

3.1	Funktionsumfang des Automic® Archive Browser v12.3.3+build1591950389463 (Abb. 2.5)	32
3.2	Benutzeranforderungen für ein relationsbasiertes Archivdaten-Recherchewerkzeug für Automic® Automation	34
3.3	Zu implementierender Funktionsumfang für ein relationsbasiertes Archivdaten- Recherchewerkzeug für Automic® Automation	36
3.4	Liste unbeliebter Programmiersprachen bei Entwicklern	38
3.5	Prozentuale Verbreitung von JDK nach Anbietern 2020	44
3.6	Prozentuale Verbreitung von JDK nach Anbietern 2021	45
3.7	Vergleich JDK Implementierungen verschiedener Anbieter	49
3.8	Fragestellungen bzgl. Dateioperationen zur Ermittlung einsetzbarer Klassen oder Methoden	55

Listings

3.1	Automic® Archive Browser Microsoft VB 5.0 Projekt	37
3.2	Ordnerstruktur von archivierten Daten	39
3.3	Ermittlung von Zeichenkodierungen archivierter Daten	41
3.4	Beispielzeile aus Indexdatei für Ausführungsdaten: UC_AH.IDX . . .	41
3.5	Überprüfung der Eindeutigkeit einer RunID innerhalb von UC_AH.IDX	41
3.6	Beispielzeile aus Indexdatei für Protokolle: UC_RH.IDX	42
3.7	Beispielzeile aus Meldungsarchiv: UC_MELD.TXT	42
3.8	Beispielzeile aus Kommentararchiv: UC_ACMT.TXT	43
4.1	ArchiveData Klassenkopf mit Attributen	59
4.2	Konstruktor der Klasse ArchiveData	59
4.3	getParents Methodenkopf	60
4.4	getRunids Methodenkopf	60
4.5	loadReport Methodenkopf	60
4.6	walkArchiveTree Methodenkopf	60
4.7	collectDataStatics Methodenkopf	61
4.8	collectDataReports Methodenkopf	61
4.9	ArchiveTreeTable Klassenkopf	61
4.10	start Methodenkopf	62
4.11	expandBranch Methodenkopf	62
4.12	searchChild Methodenkopf	62
4.13	searchAgain Methodenkopf	63
4.14	searchInTree Methodenkopf	63
4.15	isContains Methodenkopf	63
4.16	buildTaskTree Methodenkopf	63
4.17	makeTask Methodenkopf	64
4.18	newColumn Methodenkopf	64

Listings

4.19 main Methodenkopf	64
----------------------------------	----

Literaturverzeichnis

UC4 Software GmbH: *White paper Database Maintenance using the UC4 Utilities*, <https://community.broadcom.com/HigherLogic/System/DownloadDocumentFile.ashx?DocumentFileKey=adda151c-7d71-44e2-b337-7d6d1a227cb8>, 2013, letzter Zugriff: 27. 7. 2021

MarshallSoft Computing, Inc.: *Windows Standard Serial Communications for Visual Basic Programmer's Manual: Compiling Visual Basic 4/5/6 (32-bit) Programs*, http://www.marshallsoft.com/wsc_4vb.pdf, 2017, letzter Zugriff: 28. 7. 2021

ComputerWeekly.de: *Visual Basic Classic*, <https://www.computerweekly.com/de/definition/Visual-Basic-Classic>, 2016, letzter Zugriff: 27. 7. 2021

Thomas Theis: *Visual Basic .NET (Programmiersprache)*, <https://www.it-treff.de/it-lexikon/visual-basic-net-programmiersprache>, 2017, letzter Zugriff: 27. 7. 2021

Microsoft: *The .NET Language Strategy*, <https://devblogs.microsoft.com/dotnet/the-net-language-strategy/>, 2017, letzter Zugriff: 27. 7. 2021

Microsoft: *Visual Basic support planned for .NET 5.0*, <https://devblogs.microsoft.com/dotnet/the-net-language-strategy/>, 2020, letzter Zugriff: 27. 7. 2021

Dev-Insider/Stephan Augsten: *.NET 5 soll Visual Basic unterstützen*, <https://www.dev-insider.de/net-5-soll-visual-basic-unterstuetzen-a-913322/>, 2020, letzter Zugriff: 27. 7. 2021

Heise-Medien/Holger Schwichtenberg: *Gute und schlechte Nachrichten für Visual-Basic-.NET-Entwickler*, <https://www.heise.de/developer/meldung/Gute->

- [und-schlechte-Nachrichten-fuer-Visual-Basic-NET-Entwickler-4682163.html](#), 2020, letzter Zugriff: 27. 7. 2021
- Stack Exchange Inc *Stack Overflow Developer Survey 2021*, <https://insights.stackoverflow.com/survey/2021#technology-most-loved-dreaded-and-wanted>, 2021, letzter Zugriff: 27. 7. 2021
- BSI *Nachweise gemäß § 8a Absatz 3 BSIG*, https://www.bsi.bund.de/DE/Themen/KRITIS-und-regulierte-Unternehmen/Kritische-Infrastrukturen/Allgemeine-Infos-zu-KRITIS/Nachweise-erbringen/nachweise-erbringen_node.html, 2021, letzter Zugriff: 21. 7. 2021
- BaFin *Versicherungsaufsichtliche Anforderungen an die IT (VAIT) - Rundschreiben 10/2018 (VA)*, https://www.bafin.de/SharedDocs/Downloads/DE/Rundschreiben/dl_rs_1810_vait_va.pdf?__blob=publicationFile&v=5, Fassung vom 20.03.2019, letzter Zugriff: 22. 7. 2021
- BaFin Jochen Zengler Andreas Pfeßdorf *IT der Versicherer im Fokus*, https://www.bafin.de/SharedDocs/Veroeffentlichungen/DE/Fachartikel/2020/fa_bj_2010_IT_Pruefungen_VA.html, 15.10.2020, letzter Zugriff: 24. 7. 2021
- Q_PERIOR AG *Versicherungsaufsichtliche Anforderungen an die IT (VAIT)*, <https://www.q-perior.com/fokusthema/versicherungsaufsichtliche-anforderungen-an-die-it-vait/>, 2019, letzter Zugriff: 24. 7. 2021
- Wikipedia: *Versicherungsaufsichtliche Anforderungen an die IT*, https://de.wikipedia.org/wiki/Versicherungsaufsichtliche_Anforderungen_an_die_IT, 2021, letzter Zugriff: 22. 7. 2021
- VMware® Carbon Black (Hrsg.): *Global Threat Report: Extended Enterprise Under Threat*, Third Report, Juni 2020
- VMware® Carbon Black (Hrsg.): *Deutschland Threat Report: Das Extended Enterprise in Gefahr*, Dritter Report, Juni 2020
- Accenture: *State of Cyber Resilience*, https://www.accenture.com/_acnmedia/PDF-126/Accenture-State-of-Cyber-Resilience-DE.pdf#zoom=40, 2020, letzter Zugriff: 24. 7. 2021

- Göpfert, Jochen (Hrsg.): Lindenbach, Heidi (Hrsg.): *Geschäftsprozessmodellierung mit BPMN 2.0: Business Process Model and Notation*, Oldenbourg Verlag 2013
- Frank Leymann, David Schumm *Process Engine • Definition / Gabler Wirtschaftslexikon*, <https://wirtschaftslexikon.gabler.de/definition/process-engine-52692>, 2021, letzter Zugriff: 20. 7. 2021
- Freund, Jakob (Hrsg.), Rücker, Bernd (Hrsg.): „Einführung – Business Process Management“, in: *Praxishandbuch BPMN 2.0*, 4. Auflage, Carl Hanser Verlag 2014
- Freund, Jakob (Hrsg.), Rücker, Bernd (Hrsg.): „Einführung – Business Process Management“, in: *Praxishandbuch BPMN: Mit Einführung in DMN*, 6. Auflage, Carl Hanser Verlag 2019
- Thomas Zöller *Process Engine*, <https://x-integrate.com/process-engine/>, 2021, letzter Zugriff: 23. 7. 2021
- European Association of Business Process Management EABPM (Hrsg.): „Business Process Management – Definitionen“, in: *BPM CBOK® - Business Process Management BPM Common Body of Knowledge Leitfaden für das Prozessmanagement Version 3.0*, 2. Auflage, Verlag Dr. Götz Schmidt 2014
- Broadcom Inc. *Info über Automic® Automation*, https://docs.automic.com/documentation/webhelp/german/ALL/components/DOCU/12.3/Automic%20Automation%20Guides/help.htm#_Common/Welcome/AWA/About_AWA.htm, 2021, letzter Zugriff: 19. 8. 2021
- Broadcom Inc. *AEDB Archive*, https://docs.automic.com/documentation/webhelp/german/ALL/components/DOCU/12.3/Automic%20Automation%20Guides/help.htm#Utilities/admin_AboutAEDBArchive.htm, 2021, letzter Zugriff: 19. 8. 2021
- Broadcom Inc. *AE DB Reorg*, https://docs.automic.com/documentation/webhelp/german/ALL/components/DOCU/12.3/Automic%20Automation%20Guides/help.htm#Utilities/admin_AE_DB_Reorg.htm, 2021, letzter Zugriff: 19. 8. 2021
- Broadcom Inc. *AE DB Unload*, https://docs.automic.com/documentation/webhelp/german/ALL/components/DOCU/12.3/Automic%20Automation%20Guides/help.htm#Utilities/admin_AE_DB_Unload.htm, 2021, letzter Zugriff: 19. 8. 2021

- 20Guides/help.htm#Utilities/admin_AE_DB_Unload.htm, 2021, letzter Zugriff: 19. 8. 2021
- Broadcom Inc. *Archive Browser*, https://docs.automic.com/documentation/webhelp/german/ALL/components/DOCU/12.3/Automic%20Automation%20Guides/help.htm#Utilities/admin_AboutArchiveBrowser.htm, 2021, letzter Zugriff: 19. 8. 2021
- Broadcom Inc. *Serverprozesstypen*, https://docs.automic.com/documentation/webhelp/german/ALL/components/DOCU/12.3/Automic%20Automation%20Guides/help.htm#AWA/Admin/admin_types_of_server_processes.htm, 2021, letzter Zugriff: 19. 8. 2021
- Broadcom Inc. *Info über Automic® Automation - Orchestrieren von Prozessen: Workflows*, https://docs.automic.com/documentation/webhelp/german/ALL/components/DOCU/12.3/Automic%20Automation%20Guides/Content/_Common/Welcome/AWA/About_AWA.htm#link6, 2021, letzter Zugriff: 19. 8. 2021
- Broadcom Inc. *Info über Automic® Automation - Jobs*, https://docs.automic.com/documentation/webhelp/german/ALL/components/DOCU/12.3/Automic%20Automation%20Guides/Content/_Common/Welcome/AWA/About_AWA.htm#link4, 2021, letzter Zugriff: 19. 8. 2021
- Broadcom Inc. *Ausführungsphasen*, https://docs.automic.com/documentation/webhelp/german/ALL/components/DOCU/12.3/Automic%20Automation%20Guides/help.htm#AWA/Executions/obj_ExecStages_overview.htm, 2021, letzter Zugriff: 19. 8. 2021
- Jan Thomas Gundlach *Ticket 0013299: AE DB Archive (ucydbbar): Erzeugt in UC_ACMT.TXT ungültige RUNIDs*, <https://ticketing.capture.eu/view.php?id=13299>, 2021, letzter Zugriff: 19. 8. 2021
- Johannes Ahrends *Datenbank Reorganisation - Sinn oder Unsinn?*, https://www.doag.org/formes/pubfiles/2272689/114-2010-K-DB-Ahrends-Datenbank_Reorganisation-PRAESENTATION.pdf, 2010, letzter Zugriff: 20. 8. 2021
- Amtsblatt der Europäischen Union *VERORDNUNG (EU) 2016/679 DES EUROPÄISCHEN PARLAMENTS UND DES RATES*, <https://eur-lex.europa.eu/>

- [legal-content/DE/TXT/PDF/?uri=CELEX:32016R0679&from=DE](#), 4.5.2016, letzter Zugriff: 17. 7. 2021
- Bundesministerium für Wirtschaft und Energie *Europäische Datenschutz-Grundverordnung*, <https://www.bmwi.de/Redaktion/DE/Artikel/Digitale-Welt/europaeische-datenschutzgrundverordnung.html>, 2021, letzter Zugriff: 15. 7. 2021
- Niklas Plutte *Großer DSGVO-Guide mit Checkliste und Tipps für Unternehmen*, <https://www.ra-plutte.de/dsgvo-guide-checkliste-tipps-unternehmen/>, 13.07.2021, letzter Zugriff: 17. 7. 2021
- Christina Webersohn *Löschung personenbezogener Daten: wie dokumentieren?*, <https://webersohnundscholtz.de/loeschanfragen-dokumentieren/>, 31.01.2019, letzter Zugriff: 16. 7. 2021
- intersoft consulting services AG *Was ist die Datenschutz-Grundverordnung?*, <https://www.intersoft-consulting.de/infos/datenschutz-grundverordnung-dsgvo/>, 2021, letzter Zugriff: 16. 7. 2021
- AK Technik der Konferenz der unabhängigen Datenschutzbehörden des Bundes und der Länder (Hrsg.): *Das Standard-Datenschutzmodell*, https://www.datenschutzzentrum.de/uploads/sdm/SDM-Methode_V2.0a.pdf, Version 2.0a, 2019, letzter Zugriff: 17. 7. 2021
- Basil Bourque „2. Antwort“, in: Stack Exchange Inc (Hrsg.): *Difference between JVM and HotSpot?* <https://stackoverflow.com/questions/16568253/difference-between-jvm-and-hotspot/59295177#59295177>, 23.06.2021, letzter Zugriff: 29. 7. 2021
- rli „1. Antwort“, in: Stack Exchange Inc (Hrsg.): *How to implement filtering for tree-tableview* <https://stackoverflow.com/a/33216997>, 19.10.2015, letzter Zugriff: 12. 8. 2021
- Fernando Almeida *The JDKs: Which One to Use?* <https://dzone.com/articles/java-and-the-jdks-which-one-to-use>, 10.12.2018, letzter Zugriff: 05. 8. 2021
- Julius Stiebert *Neue Lizenz für Java Compatibility Kit* <https://www.golem.de/0708/54044.html>, 10.08.2007, letzter Zugriff: 05. 8. 2021

- Java Community Process Management Office: *The Java Community Process(SM) Program - Introduction - Program Overview*, <https://jcp.org/en/introduction/overview>, 2021, letzter Zugriff: 30. 7. 2021
- Oracle Corporation: *Java Client Roadmap Update - An Oracle White Paper*, <https://www.oracle.com/technetwork/java/javase/javaclientroadmapupdate2018mar-4414431.pdf>, 2018, letzter Zugriff: 24. 8. 2021
- Oracle: *Wo kann ich technische Informationen über Java erhalten?*, https://www.java.com/de/download/help/techinfo_de.html, 2021, letzter Zugriff: 30. 7. 2021
- tutego: *Die ultimativen Links zur Java 7, JDK 7 und OpenJDK 7*, <https://www.tutego.de/java/jdk7-Java-SE-7.htm>, 2011, letzter Zugriff: 30. 7. 2021
- Ziegenbalg, Michael (Hrsg.): *Objektorientierte Programmierung in Java: Software/Engineering mit Java*, BookRix GmbH & Co. KG 2018
- Hardware-Zone: *Architektur*, <https://www.hardware-zone.de/architektur.html>, 2021, letzter Zugriff: 5. 8. 2021
- Pearson Education, Informit *Java Widget Fundamentals*, <https://www.informit.com/articles/article.aspx?p=354574>, 28.01.2005, letzter Zugriff: 14. 8. 2021
- Raahim Siddiqi *Java GUI Frameworks Guide*, <https://coderslegacy.com/java-gui-frameworks-guide/>, 24.08.2020, letzter Zugriff: 14. 8. 2021
- Ullenboom, Christian (Hrsg.): *Java ist auch eine Insel*, Rheinwerk Computing 2020
- Schildt, Herbert (Hrsg.): *Java: The Complete Reference, Ninth Edition*, McGraw-Hill Education 2014
- Thorsten Thormählen *Grafikprogrammierung GUIs mit Java*, https://www.mathematik.uni-marburg.de/~thormae/lectures/graphics1/graphics_2_2_ger_web.html#3, 21.10.2019, letzter Zugriff: 15. 8. 2021
- Oracle „Is JavaFX replacing Swing as the new client UI library for Java SE?“, in: Oracle (Hrsg.): *JavaFX Frequently Asked Questions* <https://www.oracle.com/javafx/13/faq-fx.html>, 2020, letzter Zugriff: 15. 8. 2021

- [com/java/technologies/javafx/faq-javafx.html#6](https://openjfx.io/faq-javafx.html#6), 2018, letzter Zugriff: 15. 8. 2021
- OpenJFX Project *JavaFX*, <https://openjfx.io/>, 2021, letzter Zugriff: 16. 8. 2021
- Eclipse Foundation *SWT: The Standard Widget Toolkit*, <https://www.eclipse.org/swt/>, 2021, letzter Zugriff: 16. 8. 2021
- Oracle *Trail: Creating a GUI With Swing*, <https://docs.oracle.com/javase/tutorial/uiswing/>, 2021, letzter Zugriff: 15. 8. 2021
- Donald Smith *The Future of JavaFX and Other Java Client Roadmap Updates*, <https://blogs.oracle.com/java-platform-group/the-future-of-javafx-and-other-java-client-roadmap-updates>, 21.10.2019, letzter Zugriff: 15. 8. 2021
- Frederik Kirsch *JavaFX*, <https://training.cherriz.de/cherriz-training/1.0.0/oberflaechen/javafx.html>, 2021, letzter Zugriff: 15. 8. 2021
- Eclipse Contributors and others *Help Eclipse Platform: Package org.eclipse.swt.awt*, https://help.eclipse.org/2021-06/index.jsp?topic=%2Forg.eclipse.platform.doc.isv%2Freference%2Fapi%2Forg%2Fecclipse%2Fswt%2Fawt%2Fpackage-summary.html&cp%3D2_1_0_192, 2021, letzter Zugriff: 16. 8. 2021
- Eclipse Foundation, Inc. *eclipse.platform.swt.git - stats*, <https://git.eclipse.org/c/platform/eclipse.platform.swt.git/stats/?period=q&ofs=10>, 2021, letzter Zugriff: 16. 8. 2021
- GitHub, Inc. *openjdk/jfx - Commit Activity*, <https://github.com/openjdk/jfx/graphs/commit-activity>, 2021, letzter Zugriff: 16. 8. 2021
- Sven Woltmann *Dateien in Java, Teil 1: Dateien komfortabel und schnell lesen*, <https://www.happycoders.eu/de/java/dateien-einfach-schnell-lesen/>, 19.11.2019, letzter Zugriff: 18. 8. 2021
- Sven Woltmann *Dateien in Java, Teil 6: FileChannel, ByteBuffer, Memory-mapped I/O, Locks*, <https://www.happycoders.eu/de/java/strukturierte-daten-schreiben-lesen-dataoutputstream-datainputstream/>, 26.02.2020, letzter Zugriff: 20. 8. 2021

- Britta Petri, Björn Petri *Was sind Events?*, <https://www.java-tutorial.org/event-handling.html>, 2021, letzter Zugriff: 13. 8. 2021
- Eclipse Foundation, Inc. *Architektur*, <https://adoptium.net/migration.html>, 2021, letzter Zugriff: 8. 8. 2021
- Azul *OpenJDK vs Oracle JDK - Comparison Table*, <https://www.azul.com/products/core/jdk-comparison-matrix/>, 2021, letzter Zugriff: 8. 8. 2021
- New Relic (Ben Evans) *The State of Java: Trends And Data For One of the World's Most Popular Programming Languages*, <https://newrelic.com/blog/nerd-life/state-of-java>, 13.02.2020, letzter Zugriff: 8. 8. 2021
- JAXenter (Rod Cope) *The state of Java software development in 2021*, <https://jaxenter.com/java-development-2021-173870.html>, 13.02.2020, letzter Zugriff: 9. 8. 2021
- Evans, Benjamin J. (Hrsg.): Gough, James (Hrsg.): Newland, Chris (Hrsg.): *Optimizing Java: Practical Techniques for Improving JVM Application Performance*, O'Reilly Media, Inc. 2018
- [Amazon Web Services, Inc. *Amazon Corretto*, <https://aws.amazon.com/corretto/>, 24.07.2021, letzter Zugriff: 9. 8. 2021
- Eclipse Foundation, Inc. *Adoptium - Open source, prebuilt OpenJDK binaries*, <https://adoptium.net/>, 13.08.2021, letzter Zugriff: 15. 8. 2021
- Azul *Download Azul Zulu Builds of OpenJDK*, <https://www.azul.com/downloads/zulu/>, 23.07.2021, letzter Zugriff: 9. 8. 2021
- IBM *Java SDK*, <https://www.ibm.com/support/pages/java-sdk>, 29.07.2021, letzter Zugriff: 9. 8. 2021
- Pratik Patel Billy Korando *AdoptOpenJDK: An open Java distribution and community you can count on*, <https://developer.ibm.com/blogs/adoptopenjdk-an-open-java-distribution-and-community-you-can-count-on/>, 19.01.2019, letzter Zugriff: 10. 8. 2021

- Oracle Technology Network *Java SE Downloads*, <https://www.oracle.com/technetwork/java/javase/downloads/index.html>, 20.07.2021, letzter Zugriff: 9. 8. 2021
- Red Hat Developer *OpenJDK Download | Red Hat Developer*, <https://developers.redhat.com/products/openjdk/download/>, 27.07.2021, letzter Zugriff: 9. 8. 2021
- Azul *How to Keep Your Java Applications Secure*, <https://www.azul.com/blog/how-to-keep-your-java-applications-secure/>, 20.08.2020, letzter Zugriff: 9. 8. 2021
- Eclipse Foundation *Eclipse AQAvit*, <https://projects.eclipse.org/projects/adoptium.aqavit>, 2021, letzter Zugriff: 15. 8. 2021
- Eclipse Foundation *Eclipse Temurin*, <https://projects.eclipse.org/projects/adoptium.temurin>, 2021, letzter Zugriff: 15. 8. 2021
- Mark Stoodley, Samir Kamberkar *IBM joins Eclipse Adoptium and offers free certified JDKs with Eclipse OpenJ9*, <https://developer.ibm.com/blogs/ibm-joins-eclipse-adoptium-and-offers-free-certified-jdks-with-eclipse-openj9/>, 14.04.2021, letzter Zugriff: 10. 8. 2021
- Blu-ray Disc Association: *White paper Blu-ray Disc Format 2.B Audio Visual Application, Format Specifications for BD-ROM*, http://www.blu-raydisc.com/Assets/downloadablefile/2b_bdrom_audiovisualapplication_0305-12955-15269.pdf, 2005, letzter Zugriff: 1. 10. 2012
- Dooley, Wesley L. & Streicher, Ronald D.: „M–S Stereo: A Powerful Technique for Working in Stereo“, *Journ. Audio Engineering Society* vol. 30 (10), 1982
- Kuttruff, Heinrich: *Room Acoustics*, 3. Aufl., Elsevier 1991
- Spehr, Georg (Hrsg.): *Funktionale Klänge*, transcript 2009
- Sowodniok, Ulrike: „Funktionaler Stimmklang – Ein Prozess mit Nachhaltigkeit“, in: Spehr, Georg (Hrsg.): *Funktionale Klänge*, transcript 2009
- Stephenson, Uwe: „Comparison of the Mirror Image Source Method and the Sound Particle Simulation Method“, *Applied Acoustics* vol. 29, 1990

Ich versichere, die vorliegende Arbeit selbstständig ohne fremde Hilfe verfasst und keine anderen Quellen und Hilfsmittel als die angegebenen benutzt zu haben. Die aus anderen Werken wörtlich entnommenen Stellen oder dem Sinn nach entlehnten Passagen sind durch Quellenangaben eindeutig kenntlich gemacht.

Ort, Datum

Jan Thomas Gundlach