

**Visualisierung von Aufgabenrelation
innerhalb der Prozessengine
Automatic[®] Automation
Anhand archivierter Daten**

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

Thomas Gundlach
2038884



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

Erstprüfer: Prof. Dr. Larissa Putzar

Zweitprüfer: Prof. Vorname Nachname

Hamburg, 27. 8. 2021

Inhaltsverzeichnis

1	Einleitung	5
1.1	Motivation	7
1.2	Problemstellung	8
1.3	Zielsetzung	8
1.3.1	Anforderung	9
1.3.2	Abgrenzung	9
1.4	Struktur der Arbeit	9
2	Grundlagen	10
2.1	Was ist eine Prozessengine?	10
2.2	Automic® Automation	10
2.2.1	Workflows	10
2.2.2	Jobs	10
2.2.3	Runs	10
2.2.4	Reports	10
2.3	Automic® Archive Browser	10
2.4	Rechtliche Grundlagen	10
2.4.1	VAIT	10
2.4.2	ISMS / ISO 27001	10
2.4.3	EU-DSGVO	10
2.5	JDK Implementierungen	10
2.6	Java GUI-Frameworks	10
2.7	Java Dateieingangsverarbeitung	10
3	Analyse	11
3.1	Automic® Archive Browser	11
3.1.1	Mittlung von Benutzeranforderungen	11

Inhaltsverzeichnis

3.1.2	Weiterentwicklung	11
3.1.3	Datenstruktur	14
3.2	Auswahl des JDK	14
3.3	Auswahl des Java GUI-Framework	14
4	Implementierung	15
4.1	Module	15
4.2	Entwurfsmuster	15
4.3	Klassendiagramm	15
5	Zusammenfassung und Ausblick	16
5.1	Zusammenfassung	16
5.2	Ausblick	16
	Abbildungsverzeichnis	17
	Tabellenverzeichnis	19
	Listings	20
	Literaturverzeichnis	21

Abstract

Form and layout of this L^AT_EX-template incorporate the guidelines for theses in the Media Technology Department „Richtlinien zur Erstellung schriftlicher Arbeiten, vorrangig Bachelor-Thesis (BA) und Master-Thesis (MA) im Department Medientechnik in der Fakultät DMI an der HAW Hamburg“ in the version of December 6, 2012 by Prof. Wolfgang Willaschek.

The thesis should be printed single-sided (simplex). The binding correction (loss at the left aper edge due to binding) might be adjusted, according to the type of binding. This template incorporates a binding correction as BCOR=1mm (suitable for adhesive binding) in the L^AT_EX document header.

This is the english version of the opening abstract (don't forget to set L^AT_EX's language setting back to ngerman after the english text).

Zusammenfassung

Diese L^AT_EX-Vorlage berücksichtigt in Form und Layout die Vorgaben für Abschlussarbeiten im Department Medientechnik „Richtlinien zur Erstellung schriftlicher Arbeiten, vorrangig Bachelor-Thesis (BA) und Master-Thesis (MA) im Department Medientechnik in der Fakultät DMI an der HAW Hamburg“, Fassung vom 6. Dezember 2012 von Prof. Wolfgang Willaschek.

Der Ausdruck soll einseitig erfolgen (Simplex). Je nach Bindung ist ggf. die Bindekorrektur (Verlust am linken Seitenrand durch die Bindung) noch anzupassen. In dieser Vorlage ist eine Bindekorrektur im header der L^AT_EX-Datei mit BCOR=1mm für Klebebindung eingestellt.

Das ist die deutsche Version der vorangestellten Zusammenfassung. Beide Versionen – englisch und deutsch – sind verbindlich!

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 mit einander verbundene Abläufe erfolgreich stattgefunden haben. Beispielsweise müssen zur Definition von Arbeitsaufträgen Eingangsdaten wie Posteingänge aus verschiedenen Medienkanäle wie Briefpost, E-Mail oder 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 richtigen Sachbearbeiter landen. Oder immer häufiger bereits vollautomatisiert bearbeitet werden.

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. Und falls 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 stehen müssen die Datenbestände dieser Systeme regelmässig reorganisiert

1 Einleitung

werden.

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. Dass 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 unter die kritischer Infrastruktur¹ fallen können die Vorgaben restriktiv sein. Die Unternehmen oder Organisationen sind teils 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 Prozesses modelliert werden können. Für eine Überprüfung muss nachgewiesen werden, dass die Richtlinie durch die Geschäftsprozessmodellierung in einem konkreten Fall auch so wie beschrieben umgesetzt wurde. Hierfür muss die Durchführung des Geschäftsprozesses protokolliert werden und die Protokolle bei Reorganisation unverändert und unveränderbar abgelegt werden.

Im Prüfungsfall kann so theo. eine bestimmte Durchführung eines Geschäftsprozesses innerhalb des Aufbewahrungszeitraumes anhand des Protokolles nachgewiesen werden. In der Regel besteht ein Geschäftsprozess aber je nach Detaillierungsgrad aus mehreren Teilprozessen die ebenfalls aus Teilprozessen bestehen können und am Ende in einzelne Aktivitäten unterteilt sind. Jeder Geschäftsprozess, jeder Teilprozess und

¹<https://www.gesetze-im-internet.de/bsi-kritisv/BJNR095800016.html>

jede Aktivität verfügt über ein eigenes Protokoll mit entsprechendem Detaillierungsgrad. In einem komplexen Geschäftsprozess ist die Einhaltung der Vorgaben anhand des Protokolles 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 wurden. Je nach Reorganisationssystematik der Prozessengine können die Relationen an sich zwischen Geschäftsprozess, Teilprozess und Aktivität bei der Reorganisation verloren gehen. Bietet das Betrachtungswerkzeug für die reorganisierten 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 Prozessengine 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 wurden entsteht durch die Recherche von Protokollen zu Teilprozessen und Aktivitäten eines angefragten Geschäftsprozesses zur Zeit im Zweifel ein hohes Arbeitsaufkommen. Im Prüfungsfall kann die Recherche auf Verlangen des Prüfers auch während einer Prüfung und im Beisein des Prüfers durchzuführen sein. Dies wirft in der Regel kein gutes Licht auf die Nachweisbarkeit und Nachvollziehbarkeit bzgl. der Durchführungen von Geschäftsprozessen innerhalb der Prüfung.

Innerhalb der Prozessengine Automic® Automation existiert für nicht reorganisierte Protokolle die Möglichkeit zu einer Ausführung eines Geschäftsprozesses (vgl. 2.2.1) über die Relationen zu Teilprozessen bis hin zu einer Aktivität (vgl. 2.2.2) zu traversieren. Dies ermöglicht eine deutlich übersichtlichere und schnellere Recherche durch die Protokolle komplexer Geschäftsprozesse. Daraus entstand der Anreiz die Recherchemöglichkeiten der Prozessengine Automic® Automation und des Betrachtungswerkzeuges für reorganisierte Protokolle zu harmonisieren.

1.2 Problemstellung

Der Automic® Archive Browser, das bestehende Betrachtungswerkzeug für reorganisierte Protokolle der Prozessengine Automic® Automation, ist für eine flachen Recherche ausgelegt. Die dem zu Grunde liegenden Datenstrukturen der reorganisierten Protokolle sind entsprechend aufgebaut und nicht dokumentiert. Auch bietet der Automic® Archive Browser 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. Zum anderen 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 reorganisierte Protokolle anhand von Automic® Automation vorgestellt. Vordergründig wird die grundlegende Entwicklung des Betrachtungswerkzeuges durchgeführt.(vgl. [Sowodniok 2009](#)).

1.3 Zielsetzung

Ziel der Arbeit ist eine relationsorientierte Darstellung von reorganisierten Ausführungsprotokollen und die Implementierung der dafür notwendigen Grundfunktionalitäten. Hierzu gehört die Auswahl eines passenden Frameworks und das Erarbeiten einer entsprechenden Architektur. Besonderen Wert wird auf die Benutzerfreundlichkeit gelegt unter Berücksichtigung des angewendeten Frameworks. Zudem soll auf

Grundlage 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. Davon ausgehend ist die technische Grundlage für ein relationsorientiertes Recherchewerkzeug für reorganisierte 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 reorganisierten Daten, die Darstellung der Relationen der reorganisierten Daten sowie die Möglichkeit nach bestimmten Informationen zu filtern.

1.3.2 Abgrenzung

Diese Arbeit ist explorativ. Der Fokus liegt primär auf der Struktur der reorganisierten Daten der Prozessengine Automic[®] Automation. Es können innerhalb des zeitlich begrenzten Rahmens der Bachelorarbeit keine umfangreichen Benutzer- oder Systemtests durchgeführt werden. Grafisch werden die Benutzerschnittstellen wie auch sämtliche Fensterkomponenten im Standardstil des ermittelten GUI-Frameworks angezeigt.

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 eine Prozessengine?

2.2 Automic[®] Automation

2.2.1 Workflows

2.2.2 Jobs

2.2.3 Runs

2.2.4 Reports

2.3 Automic[®] Archive Browser

2.4 Rechtliche Grundlagen

2.4.1 VAIT

2.4.2 ISMS / ISO 27001

2.4.3 EU-DSGVO

2.5 JDK Implementierungen

2.6 Java GUI-Frameworks

2.7 Java Dateieingangsverarbeitung

3 Analyse

3.1 Automic® Archive Browser

3.1.1 Mittlung von Benutzeranforderungen

Funktionsumfang

Benutzeranforderungen

Funktionsumfang Implementierung

3.1.2 Weiterentwicklung

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

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. [Wikipedia 2021](#)). Entsprechend stehen keine Sprachmittel aus dem .NET-Framework zur Verfügung. Für den Nachfolger des auf dem .NET-Framework basierenden VB .NET (vgl. [Wikipedia 2021](#)) 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 seit dem 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.

Gefördert wurde diese Entwicklung durch Nutzer anderer Hochsprache 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

3 Analyse

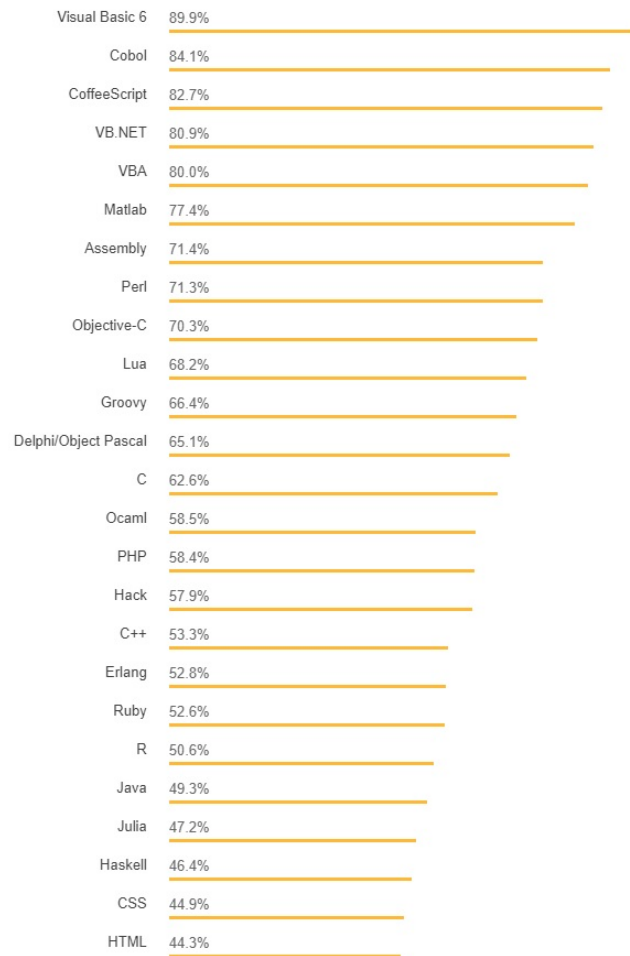


Abbildung 3.1: Liste der Programmiersprachen die Entwickler lieber nicht einsetzen¹

Ruf (vgl. [Heise-Medien 2020](#)). Das die Ökosysteme, die verbreitete Syntax und der Umfang von Sprachen wie Java oder C# inzwischen VB weitausüberlegen sind wird am Stack Overflow Developer Survey von 2018 (Abb. 3.1) deutlich. In diesem führte Visual Basic die Liste der Sprache an, die Nutzer ungerne einsetzen - Zugleich wurde 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.

¹Quelle: Stackoverflow Developer Survey Results 2018 https://insights.stackoverflow.com/survey/2018#technology_-_most-loved-dreaded-and-wanted-languages Aufruf: 28.07.2021

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

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

some text more and other text

3.2 Auswahl des JDK

some textmore and other text

3.3 Auswahl des Java GUI-Framework

some textmore and other textmore and other text

4 Implementierung

some textmore and other textmore and other textmore and other textmore and other
textmore and other textmore and other textmore and other textmore and other text-
more and other text

4.1 Module

4.2 Entwurfsmuster

4.3 Klassendiagramm

5 Zusammenfassung und Ausblick

5.1 Zusammenfassung

5.2 Ausblick

Abbildungsverzeichnis

3.1 Liste der Programmiersprachen die Entwickler 2018 lieber nicht einsetzen 13

Abkürzungsverzeichnis

VB Visual Basic

VBA Visual Basic for Applications

Tabellenverzeichnis

Listings

3.1	Automic® Archive Browser Microsoft VB 5.0 Projekt	12
-----	---	----

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

Wikipedia: *Visual Basic Classic*, https://de.wikipedia.org/wiki/Visual_Basic_Classic, 2021, letzter Zugriff: 27. 7. 2021

Wikipedia: *Visual Basic .NET*, https://de.wikipedia.org/wiki/Visual_Basic_.NET, 2021, 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.

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

Thomas Gundlach