GI-Edition

Lecture Notes in Informatics

Gesellschaft für Informatik (Hrsg.)

SKILL 2018

Studierendenkonferenz Informatik

26. und 27. September 2018
Berlin

Gesellschaft für Informatik (Hrsg.): Studierendenkonferenz 2018

14

Seminars

Gesellschaft für Informatik (Hrsg.)

SKILL – Studierendenkonferenz Informatik 2018

26./ 27. September 2018
Berlin

Gesellschaft für Informatik e.V. (GI)

Lecture Notes in Informatics (LNI) - Seminars
Series of the Gesellschaft für Informatik (GI)

Volume S 14

ISSN 1614-3213
ISBN 978-3-88579-448-6

Volume Editors
Gesellschaft für Informatik e.V.
Ahrstraße 45
53175 Bonn
E-Mail: gs@gi.de
Redaktion: Michael Becker
E-Mail: mbecker@informatik.uni-leipzig.de

printed by Köllen Druck+Verlag GmbH, Bonn

Vorwort

Die Studierendenkonferenz Informatik (SKILL) ist eine jährliche Konferenz für Studentinnen und Studenten der Informatik sowie angrenzender Disziplinen aus ganz Deutschland. Die Intension dieses Konferenzformates ist es, sehr guten studentischen Arbeiten eine öffentliche Plattform zur Diskussion zu bieten. Die Studierenden können Erfahrungen zum wissenschaftlichen Publizieren sammeln und ihre Ergebnisse vor einem breiten Publikum vorstellen.

In diesem Jahr wurden insgesamt 29 Beiträge als Full oder Short Paper eingereicht und wissenschaftlich begutachtet. In diesem Band erscheinen 15 Beiträge, die auf zwei Konferenztagen durch die Studierenden präsentiert wurden. Sie bilden eine breite Vielfalt an Themen und aktuellen wissenschaftlichen Fragestellungen ab. Die Einreichungen dieses Jahres weisen einen besonderen Schwerpunkt mit Bezug zu neuronalen Netzwerken, insbesondere Convolutional Neural Networks, auf.

Mit der SKILL 2018 wurde die seit 2014 bestehende Zusammenarbeit mit der Gesellschaft für Informatik e.V. (GI) weiter vertieft. Die Konferenz fand am 26. und 27. September 2018 im Rahmen der größten deutschsprachigen Informatik-Konferenz, der INFORMATIK 2018 Jahrestagung der GI, in Berlin statt.

Die Mitglieder des Organisationskomitees der SKILL 2018 bedanken sich zunächst bei den Autorinnen und Autoren, ohne deren qualitativ hochwertige Beiträge die Konferenz nicht möglich wäre. Das Themenspektrum eingereichter Beiträge reicht dabei von theoretisch-mathematischen Grundlagenarbeiten über Beiträge zur technischen Informatik bis hin zur Nutzung von Erkenntnissen der Informatik in anderen Fachbereichen.

Wir freuen uns darüber hinaus, dass wir auch in diesem Jahr wieder namhafte Gutachterinnen und Gutachter gewinnen konnten, die den Studierenden mit hilfreichen und ausführlichen Kommentaren zu ihren Arbeiten zur Seite standen. Durch ihre Hilfe konnten alle eingereichten Beiträge von jeweils zwei ausgewiesenen Expertinnen und Experten begutachtet werden.

Die Mitglieder des Organisationskomitees bedanken sich für das große Interesse an der SKILL und zwei spannende Konferenztage.

Berlin, 26. September 2018

Organisationskomitee der SKILL 2018

- Michael Becker, Institut für Angewandte Informatik e.V.
- Judith Michael, Rheinisch-Westfälische Technische Hochschule Aachen
- Thomas Riechert, Hochschule für Technik, Wirtschaft und Kultur Leipzig
- Johannes Schmidt, Universität Leipzig

Gutachterinnen und Gutachter der SKILL 2018

- Erika Abraham, RWTH Aachen
- Georges Alkhouri, ScaDS Dresden/Leipzig
- Ernst Althaus, Johannes Gutenberg Universität Mainz
- Matthias Book, University of Iceland
- Hans-Joachim Bungartz, Technische Universität München
- Markus Dahm, Hochschule Düsseldorf
- Addis Dittebrandt, Karlsruhe Institut für Technologie
- Herbert Fischer, Technische Hochschule Deggendorf
- Felix Freiling, Friedrich-Alexander-Universität Erlangen-Nürnberg
- Julia Friedrich, Institut für Angewandte Informatik e.V., Leipzig
- Kurt Geihs, Universität Kassel
- Walter Hower, Hochschule Albstadt-Sigmaringen
- Paul Jähne, Fraunhofer-Institut für Zelltherapie und Immunologie
- Christine Jakobs, Technische Universität Chemnitz
- Nils Jansen, Radboud University Nijmegen
- Enkelejda Kasneci, Eberhard Karls Universität Tübingen
- Friedbert Kaspar, Hochschule Furtwangen
- Oliver Keszöcze, Universität Bremen
- Stephan Klingner, Institut für Angewandte Informatik e.V.
- Michael König, Karlsruhe Institut für Technologie
- Christian Kücherer, Universität Heidelberg
- André Langer, Technische Universität Chemnitz
- Isabel Leber, Hochschule Reutlingen
- Jan-Patrick Lehr, Graduate School of Computational Engineering at Technische Universität Darmstadt
- Mathias Lux, Alpen-Adria-Universität Klagenfurt
- Ludger Martin, Hochschule RheinMain Wiesbaden
- Alexander Mehler, Goethe-Universität Frankfurt am Main
- Kyrill Meyer, Institut für Angewandte Informatik e.V.
- Klaus Meyer-Wegener, Friedrich-Alexander-Universität Erlangen-Nürnberg
- Felix Neumeister, Karlsruhe Institut für Technologie
- Axel Ngonga, Paderborn University
- Jonas Oppenländer, Freie Universität Berlin
- Sabine Radomsi, Hochschule für Telekommunikation Leipzig
- Stefan Rass, Alpen-Adria-Universität Klagenfurt
- David Georg Reichelt, Universität Leipzig
- Petra Sauer, Beuth Hochschule für Technik Berlin
- Peter Schartner, Alpen-Adria-Universität Klagenfurt
- Thomas Schmid, Universität Leipzig
- Andreas Schmidt, Universität des Saarlandes (Saarbrücken)
- Ingo Scholtes, ETH Zürich
- Oliver Skroch, Hochschule Darmstadt
- Detlef Stern, Hochschule Heilbronn
- Klaus Volbert, Ostbayerische Technische Hochschule Regensburg
- Ralf Wimmer, Albert-Ludwigs-Universität Freiburg

Inhaltsverzeichnis

Informatik Grundlagen

Multimedia und Datenverarbeitung

Business IT

Informatik in der Anwendung

Neuronale Netze

Autorenverzeichnis

Informatik Grundlagen

OIDC-Agent: Managing OpenID Connect Tokens on the Command Line

Gabriel Zachmann [1,2]

Abstract: OpenID Connect is widely used in Authentication and Authorization Infrastructures including the infrastructures of multiple EU projects like INDIGO-DataCloud, the Human Brain Project or the European Open Science Cloud. Due to their nature, OpenID Connect Access Tokens are currently not straightforward to use from the command line. They have a high character count and are short lived. Therefore, they de facto have to be copied from a source providing the access token, most likely a web service. Considering this insufficient usability from the command line, our goal was to overcome this by developing a tool to manage OpenID Connect tokens. We present the design of this tool named `oidc-agent` and possible usages. The design is oriented at the `ssh-agent`, providing the user a familiar way to handle OpenID Connect tokens. By splitting the whole service into multiple components we also ensure privilege separation. We implemented a daemon to manage OpenID Connect tokens (`oidc-agent`), a tool for generating agent account configurations (`oidc-gen`) and a tool for loading and unloading these configurations from the agent (`oidc-add`). Additionally, we provide application programming interfaces for agent clients through C and UNIX domain sockets. We also provide an example agent client (`oidc-token`) that can be used to easily get an access token from `oidc-agent` using the command line. Therefore, users do not need to handle long, unhandy access tokens, but the application can obtain an access-token through `oidc-agent` when needed. All components can be freely used and are available on GitHub[3] under the MIT license.

Keywords: OpenID Connect; OIDC; oidc-agent; authorization; authentication; security; command line

1 Introduction

OpenID Connect (OIDC) [Sa14] is an authentication protocol based on OAuth2 [Ha12] and an important key component in many modern Authentication and Authorization Infrastructures (AAIs) including those of several EU projects. At the Karlsruhe Institute of Technology (KIT) we are engaged in multiple of these projects in the field of AAI. Users and developers of these projects and other AAIs have to handle OIDC access tokens on a regular base. While for web applications this is easy to do, handling OIDC access tokens on the command line is currently a cumbersome and repetitive procedure requiring manual copying and pasting.

[1] Karlsruhe Institute of Technology, Steinbuch Centre for Computing, Herrmann-von-Helmholtz-Platz 1, 76344 Eggenstein-Leopoldshafen, Germany gzachmann@outlook.com
[2] Baden-Wuerttemberg Cooperative State University Karlsruhe, Germany
[3] https://github.com/indigo-dc/oidc-agent

Currently there is no other tool for managing OIDC tokens on the command line. However, there are command line tools for dynamically registering OIDC clients [Go15; Sk17], but these tools are only for registering and managing clients and not for handling access tokens. Therefore, our goal was to develop an architecture for a tool that can do both: registering clients and handling OIDC tokens. By implementing the developed architecture we wanted to integrate OIDC with the command line with just one tool chain.

2 Architecture

While developing the architecture of `oidc-agent` [IN18] we followed the `ssh-agent` [Op17] design, because `oidc-agent` has a similar purpose as the `ssh-agent`, but for OIDC tokens instead of ssh keys. By following the `ssh-agent` design, users are able to use `oidc-agent` in a way they are used to from `ssh-agent`.

The architecture consists of multiple components:

`oidc-agent`: The actual agent managing the tokens and performing all communication with the OpenID Provider (OP).

`oidc-gen`: A tool for generating account configuration files for usage with `oidc-agent` and `oidc-add`.

`oidc-add`: A tool that loads the account configurations into `oidc-agent`.

`oidc-token` and third party applications: Applications that need an OIDC access token can obtain it through the agent's application programming interface (API). One example application for getting an access token is `oidc-token`.

We will describe these components in the following subsections. The architecture of `oidc-agent` is visualized in Figure 1.

2.1 `oidc-agent`

`oidc-agent` is the central component of the `oidc-agent` project. It runs as a daemon in the background and handles all communication with the OPs. Other applications (including `oidc-gen`, `oidc-add`, and `oidc-token`) have to use an UNIX Domain Socket to communicate with the agent. To locate the socket an environment variable is used. This variable has to be set when starting the `oidc-agent`. The required shell command is printed by `oidc-agent` on startup. This is the same process as it is for `ssh-agent` [Ma16]. The access control for the used socket is handled by the file system. Therefore, any process started by the same user can communicate with the agent.

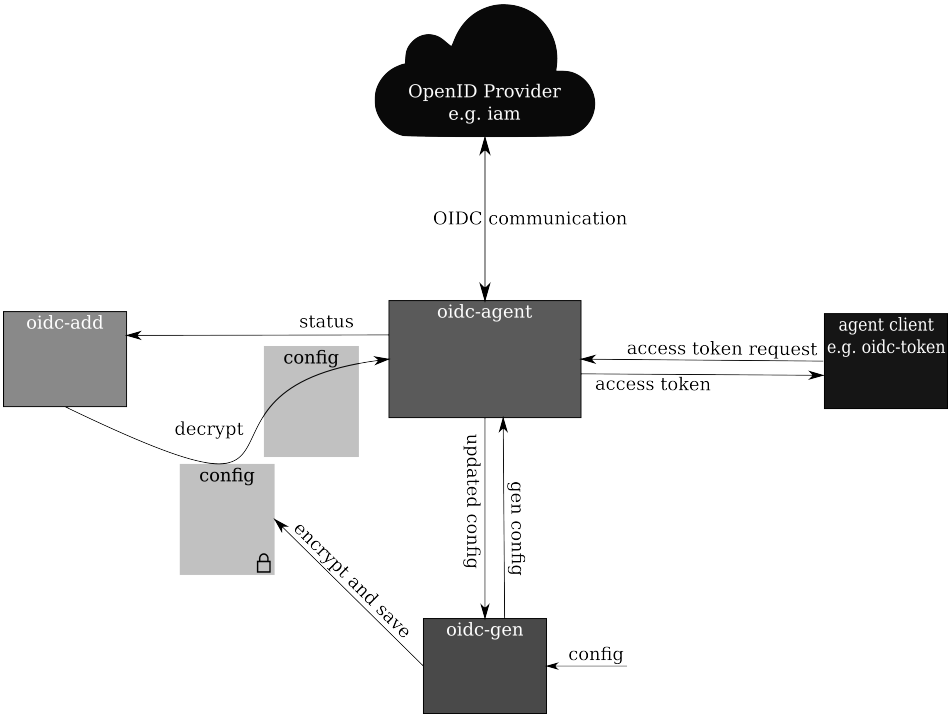


Fig. 1: Architectural Design of `oidc-agent`

2.2 `oidc-gen`

`oidc-gen` is used to generate account configuration files. To use `oidc-agent` with an OP an account configuration is needed. To be able to obtain access tokens `oidc-agent` needs a registered OIDC client. Some OPs support dynamic client registration which can be used by a client to register itself with the OP. For these providers dynamic client registration can be used by `oidc-gen` and `oidc-agent` to register the client and then generate the account configuration file. If the OP does not support dynamic client registration, users have to register a client themselves and then provide the relevant information to `oidc-gen` to create the configuration file.

2.3 `oidc-add`

`oidc-add` is used to add an account configuration to the agent. `oidc-add` will read the encrypted account configuration file, decrypt it with the user supplied password and send the configuration to `oidc-agent`. After adding a configuration an access token can be obtained

for this configuration (e.g. by using `oidc-token`). `oidc-add` can also be used to remove an already loaded configuration from the agent.

2.4 `oidc-token`

`oidc-token` is an example application that is able to obtain an access token through `oidc-agent`. Other applications that need to get an access token can use the provided API. There are two ways of using the API. We provide a C-API that hides the communication details under simple function calls. Therefore, other applications do not have to handle locating, writing and reading of the UNIX domain socket. Many languages support calling C functions which makes it easy to integrate in any application. The C-API can be used by including the source files or by using the provided library. `oidc-token` uses this C-API and can be used as a reference implementation on how to use this API.

If an application cannot or does not want to call C functions, it can communicate directly with `oidc-agent` through the UNIX Domain Socket. To do so, the application has to obtain the socket path from the environment and connect to it. Socket communication is done through JavaScript Object Notation (JSON) encoded messages.

2.5 Privilege Separation

By following the security by design principles and splitting the system's functionalities into multiple components we also achieved privilege separation. Table 1 shows the privileges every component needs. We emphasize that the only file `oidc-agent` reads is the Certificate Authority (CA) bundle file needed for Transport Layer Security (TLS) encrypted communication. So the agent - as the only component that has network access - does not access disk (with the CA bundle file as an exception). Please also note that `oidc-gen` does not need to execute any files to work correctly. However, it needs the execution right for automatically opening the authorization uniform resource locater (URL) in a web browser when performing the Authorization Code Flow.

component	ipc	network	read file	write file	execute file
`oidc-agent`	✓	✓	(✓)	✗	✗
`oidc-gen`	✓	✗	✓	✓	(✓)
`oidc-add`	✓	✗	✓	✗	✗
agent clients	✓	✗	✗	✗	✗

Tab. 1: Privilege Separation in the oidc-agent project

2.6 Usage of OpenID Connect Flows

As already mentioned, `oidc-agent` is able to use dynamic client registration to automatically register an OIDC client, but `oidc-agent` supports more OIDC flows. Normally, the agent uses the Refresh Flow to obtain an access token using a refresh token. This flow is illustrated in Figure 2. The used refresh token is stored in an encrypted way and is obtained when the account configuration is generated. `oidc-agent` supports multiple ways to obtain this refresh token.

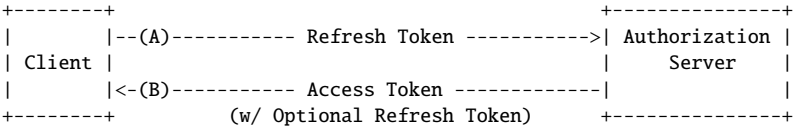


Fig. 2: Refreshing an Expired Access Token. Adapted from [Ha12]

- **Out of Band**: The refresh token can be obtained out of band and be provided directly. But it is very unlikely that a user is able to obtain a refresh token for the used client, because refresh tokens are bound to a specific client id. This functionality is included for development and for potential advanced use cases.
- **Password Flow**: The Resource Owner Password Credentials can be used directly as an authorization to obtain a refresh (and access) token [Ha12]. Obviously this flow reveals the user's credentials to `oidc-agent`. We emphasize that `oidc-agent` does not store this information and that it is held in memory as short as possible, but users might want to use one of the other flows that do not reveal the user's credentials to `oidc-agent`. However, this is the only flow that can be done entirely on the command line; the other flows require some sort of web-based authentication. This flow is visualized in Figure 3.
- **Authorization Code Flow**: The Authorization Code Flow is the most widely spread OAuth2 / OIDC flow and is the standard web flow. No user credentials are revealed to `oidc-agent`, instead the user authenticates against the OP using a web browser. This flow requires `oidc-agent` to start a small web server to receive the OP's response. Out of the implemented flows this is the one mostly supported by OPs; thus it is also the default flow for `oidc-agent`. Figure 4 shows how the authorization code flow works.
- **Device Flow**: The Device Flow is an OAuth2 flow specially for browserless and input constrained devices [De17]. It uses a second device to perform the authentication against the OP in a browser and in that way also does not reveal the credentials to `oidc-agent`. Because the authentication is done on a second device, there is no web interaction needed on the primary device, i.e. on the primary device it can be done using only the command line. The flow is illustrated in Figure 5.

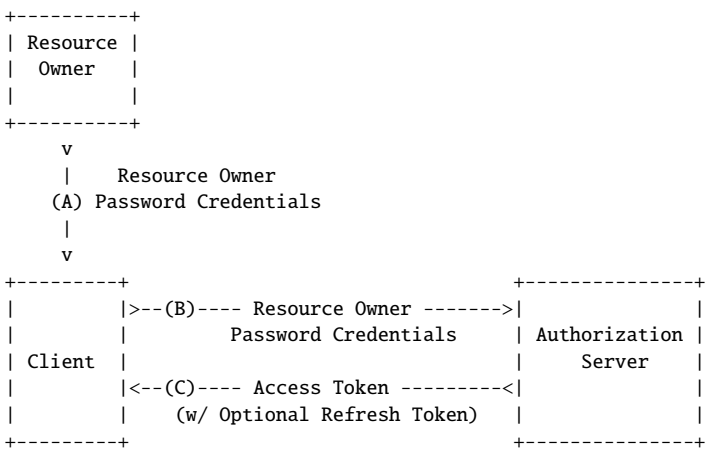


Fig. 3: Resource Owner Password Credentials Flow [Ha12]

The Authorization Code Flow is clearly the most used flow to obtain a refresh token in `oidc-agent`, because it is supported by almost any OP and because the password flow and device flow are supported by only few OPs. Please note that the device flow specification is currently not in the state of being an RFC but only an Internet Engineering Task Force (IETF) draft.

2.7 General Principles

Because `oidc-agent` is a security related software that handles sensitive data, we have to handle it with the necessary caution. As already mentioned, user credentials are not stored on disk and kept in memory as short as possible. To ensure this, we clear all allocated memory before freeing it. By clearing all allocated memory and not only sensitive data we ensure that we do not accidentally leak information (e.g. a server response that contains a refresh token as a sub-string).

The stored data contains sensitive elements - in particular the refresh token and client secret. All data written to disk is therefore encrypted using the easy to use, cross-platform encryption library `libsodium` [17a]. The used encryption algorithm is `XSALSA20`. It is a stream cipher based on `SALSA20` but with a 192 bits long nonce instead of 64 bits [17b]. XSalsa20 uses a 256-bit key, that is derived from the users's encryption password, as well as the first 128 bits of the randomly generated nonce to compute a subkey [17b]. This subkey and the remaining 64 bits of the nonce are the parameters for the `Salsa20` function used to eventually generate the stream [17b].

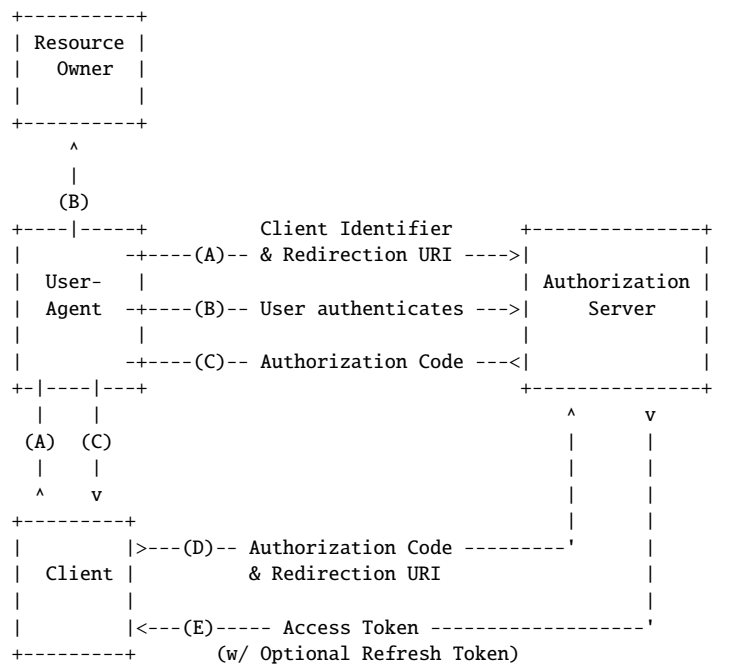


Fig. 4: Authorization Code Flow [Ha12]

3 Use Cases

Usages of `oidc-agent` can be categorized into two groups. An application can utilize `oidc-agent` to obtain an access token or the user can obtain the access token from the agent and provide it to an application. For both of these groups we will describe possible use cases.

3.1 Applications Utilizing `oidc-agent`

Using an application that utilizes `oidc-agent` to obtain an access token is very easy. Instead of manually obtaining an access token and providing it to the application (e.g. through an environment variable) the application has to be modified to support `oidc-agent` natively. The application then only needs the name of the account configuration to be used. In addition, `oidc-agent` has to be running and the account configuration has be loaded, i.e. by using `oidc-add`. Then the application can be used as normal.

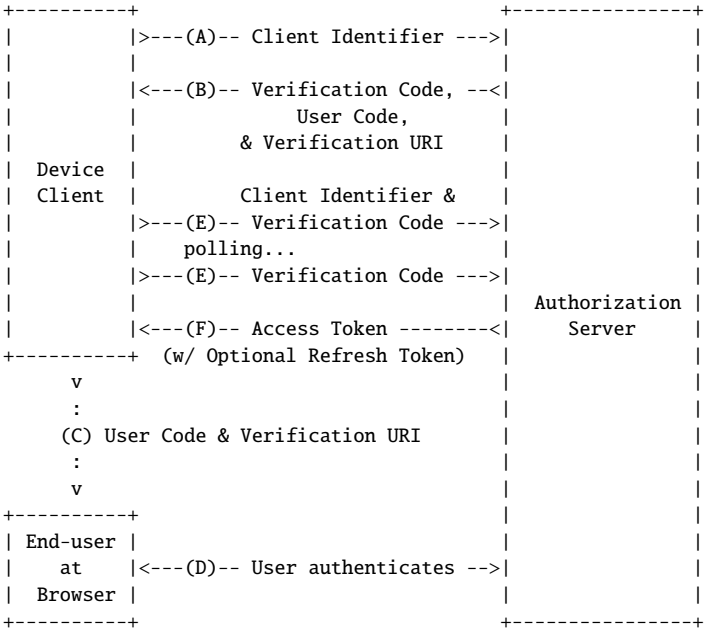


Fig. 5: Device Flow [De17]

An example of an application that can utilize `oidc-agent` to obtain access tokens is `wattson`. `wattson` [IN17b] is a command line client for `WaTTS` [IN17a], the INDIGO [18d] Token Translation Service (TTS) that translates OIDC credentials to non-OIDC credentials (e.g. an X.509 certificate). An example on how to use `wattson` with `oidc-agent` is shown in Listing 1. Lines 1–3 initialize the environment variables needed by `wattson`; lines 4–8 startup `oidc-agent` and load the "indigo-iam" account configuration; line 10–16 show the actual `wattson` call and the resulting output.

3.2 Using `oidc-agent` to Obtain an Access Token

If an application does not use our API, but needs an OIDC access token, `oidc-token` can be used to obtain the access token from `oidc-agent`. Most likely the application will read the access token from an environment variable. Setting this variable with `oidc-token` is very straightforward as can be seen from Listing 2. Again lines 1–6 show the agent startup and the loading of the account configuration. In line 7 the environment variable is filled with the requested access token; after that it can be used by the application.

```
user@oidc:~$ export WATTSON_URL=https://watts.data.kit.edu
user@oidc:~$ export WATTSON_ISSUER=iam
user@oidc:~$ export WATTSON_AGENT_ACCOUNT=indigo-iam
user@oidc:~$ eval `oidc-agent`
Agent pid 2018
user@oidc:~$ oidc-add indigo-iam
Enter encrpytion password for account config indigo-iam:
success

user@oidc:~$ wattson request x509
connecting to https://watts.data.kit.edu/api/v2/iam/ using protocol version 2
received token from oidc-agent
requesting credential for service [x509]:
Credential [a2fa1b39-8723-4385-aae1-6d768a5a2f88]:
[ Certificate (textfile)] => Certificate:
...
```

List. 1: Using `wattson` with `oidc-agent`

```
user@oidc:~$ eval `oidc-agent`
Agent pid 2018
user@oidc:~$ oidc-add indigo-iam
Enter encrpytion password for account config indigo-iam:
success

user@oidc:~$ export WATTSON_TOKEN=`oidc-token indigo-iam`
```

List. 2: Using `oidc-token` to pass an access token through an environment variable

Of course `oidc-token` may as well be used to get the access token and then use it in any other way necessary. To obtain an access token from `oidc-agent` with `oidc-token` a simple call to `oidc-token` providing the account configuration name is enough.

One scenario in which this access token might be used is inside the Human Brain Project (HBP). The HBP [18c] is a Future and Emerging Technologies (FET) flagship project as part of the Horizon 2020 programme [Eu] of the European Union (EU) and aims at a better understanding of the human brain. Users of the HBP can use a High Performance Computing (HPC) system consisting of supercomputing sites in Jülich, Barcelona, Bologna and Lugano and a cloud storage system in Karlsruhe [Be16]. This HPC system can be accessed through the UNiform Interface to COmputing REsources (UNICORE) [18e] by using OIDC. This means that `oidc-agent` can be used to get an access token for accessing the HPC system.

4 Conclusion

With `oidc-agent` we developed a tool for managing OIDC access tokens on the command line. The architecture is based on `ssh-agent` and provides the user a way to handle OIDC tokens similar to ssh keys. The `oidc-agent` architecture also supports privilege separation and secure memory deallocation.

Users can easily generate new account configurations with `oidc-gen`. After the account configuration is loaded into `oidc-agent` using `oidc-add`, an application that needs an access token can request it using the API. Additionally `oidc-token` can be used to get an access token on the command line.

`oidc-agent` supports several OIDC flows like the password flow, authorization code flow and device flow. This allows `oidc-agent` to be widely used with many different OPs. We verified our implementation with B2Access [EU16], EGI-Checkin [18b], Elixir [18a], Google [Go], HBP and INDIGO IAM.

`oidc-agent` is released under the MIT license as part of the INDIGO-Datacloud project. Source code and releases are available at GitHub[4]. To improve the installation process we want to add the oidc-agent package into the repositories of the major linux distributions.

Acknowledgments

This project was done has part of my dual study at the Baden-Wuerttemberg Cooperative State University Karlsruhe within the practical phase at the KIT-SCC.

I want to thank my supervisors Bas Wegh and Marcus Hardt for their feedback, guidence, and advise throughout the project.

Additionally, I want to thank the anonymous reviewers for their comprehensive feedback.

References

[17a] libsodium | A modern, portable, easy to use crypto library, 2017, URL: `https://github.com/jedisct1/libsodium`.

[17b] The Sodium crypto library (libsodium), 1.0.13, libsodium, July 2017, URL: `https://www.gitbook.com/book/jedisct1/libsodium/details`, visited on:

[18a] ELIXIR | A distributed infrastructure for life-science information, 2018, URL: `https://www.elixir-europe.org/`.

[18b] European Grid Infrastructure, 2018, URL: `https://www.egi.eu/`.

[4] https://github.com/indigo-dc/oidc-agent

[18c] Human Brain Project, 2018, URL: https://www.humanbrainproject.eu/.

[18d] INDIGO DataCloud, 2018, URL: https://www.indigo-datacloud.eu/.

[18e] UNICORE | Distributed computing and data resources, 2018, URL: https://www.unicore.eu/.

[Be16] Benedyczak, K.; Schuller, B.; Sayed, M. P.-E.; Rybicki, J.; Grunzke, R.: UNICORE 7 - Middleware Services for Distributed and Federated Computing. In: 2016 International Conference on High Performance Computing Simulation (HPCS). Pp. 613–620, July 2016.

[De17] Denniss, W.; Google; Bradley, J.; Identity, P.; Jones, M.; Microsoft; Tschofenig, H.; Limited, A.: OAuth 2.0 Device Flow for Browserless and Input Constrained Devices, tech. rep., Oct. 2017, URL: https://tools.ietf.org/html/draft-ietf-oauth-device-flow-07.

[Eu] European Commission: Horizon 2020 - European Commission, URL: https://ec.europa.eu/programmes/horizon2020/.

[EU16] EUDAT: B2ACCESS - EUDAT, Nov. 2016, URL: https://www.eudat.eu/services/b2access.

[Go] Google: Google Identity Platform, URL: https://developers.google.com/identity/.

[Go15] Goering, B.: openid-cli: A CLI for interacting with an OpenID Connect provider, 2015, URL: https://github.com/gobengo/oidc-cli,%20visited%20on:%20June%206,%202018.

[Ha12] Hardt, D.: The OAuth 2.0 Authorization Framework, RFC 6749, RFC Editor, Oct. 2012, URL: http://www.rfc-editor.org/info/rfc6749.

[IN17a] INDIGO-Datacloud: WaTTs - the INDIGO Token Translation Service, 2017, URL: https://github.com/indigo-dc/tts.

[IN17b] INDIGO-Datacloud: wattson: the watts command line client, 2017, URL: https://github.com/indigo-dc/wattson.

[IN18] INDIGO-Datacloud: oidc-agent for managing OpenID Connect tokens, 2018, URL: https://github.com/indigo-dc/oidc-agent.

[Ma16] Manual, L. P.: SSH-AGENT(1) BSD General Commands Manual, Nov. 2016, URL: http://man7.org/linux/man-pages/man1/ssh-agent.1.html.

[Op17] OpenBSD: OpenSSH, 2017, URL: https://www.openssh.com/.

[Sa14] Sakimura, N.; Bradley, J.; Jones, M.; de Medeiros, B.; Mortimore, C.: OpenID Connect Core 1.0 incorporating errata set 1, tech. rep., Nov. 2014, URL: http://openid.net/specs/openid-connect-core-1_0.html.

[Sk17] Skokan, F.: openid-client-cli: CLI for managing dynamic OpenID Connect client registrations. 2017, URL: https://github.com/panva/openid-client-cli.

Wissensrevision in der reaktiven Antwortmengenprogrammierung

Jonas Philipp Haldimann[1]

Abstract: Reaktive Antwortmengenprogrammierung [Ge11; Ge15] ist eine neuere Erweiterung der Antwortmengenprogrammierung, welche sich wiederum für Planungs- und andere Suchprobleme verwenden lässt. Die Erweiterung zur reaktiven Antwortmengenprogrammierung ermöglicht es, Antwortmengenprogramme zu ergänzen oder zu ändern, noch nachdem sie bereits grundiert wurden. Dadurch lässt sie sich auch in dynamischen Umgebungen effizient einsetzen.

Eine solche Anpassung ist eine Form der Wissensrevision. Diese Revision soll hier für Multi-Shot Solver [Ge15], eine Form der reaktiven Antwortmengenprogrammierung, genauer untersucht werden. Dazu werden wir zunächst die AGM-Kriterien [AGM85], die häufig für die Untersuchung der Revision auf Mengen logischer Aussagen verwendet werden, an die reaktive Antwortmengenprogrammierung anpassen. Anschließend untersuchen wir die Revision bei der reaktiven Antwortmegenprogrammierung mit den angepassten AGM-Kriterien und den Basisrevisionskriterien aus [KK12].

Keywords: (Reaktive) Antwortmengenprogrammierung; Wissensrevision; AGM-Theorie; Basisrevisionskriterien

1 Einleitung

Reaktive Antwortmengenprogrammierung [Ge11; Ge15] ist eine neuere Erweiterung der Antwortmengenprogrammierung. Diese wiederum ist eine Form der logischen Programmierung, die auf der Suche nach Antwortmengen zu gegebenen Programmen basiert. Antwortmengenprogrammierung lässt sich für Planungs- und andere Suchprobleme verwenden. Die Erweiterung zur reaktiven Antwortmengenprogrammierung ermöglicht es, Antwortmengenprogramme zu ergänzen oder zu ändern, noch nachdem sie bereits grundiert wurden. Dadurch lässt sich Antwortmengenprogrammierung auch in dynamischen Umgebungen effizient einsetzen.

Betrachten wir beispielsweise einen Roboter, der Kisten in einer Lagerhalle transportiert. Wir wollen Antwortmengenprogrammierung nutzen um die Aktionen des Roboters zu planen. Die Regeln, welche die Bewegung des Roboters beschreiben, bleiben konstant. Einige Informationen, wie der Startpunkt des Roboters oder die abzuarbeitenden Aufträge, ändern sich jedoch. Reaktive Antwortmengenprorammierung ermöglicht es, diese Informationen effizient zur Laufzeit anzupassen.

[1] TU Dortmund, Fakultät für Informatik, Deutschland; Kontakt: jonas.haldimann@tu-dortmund.de

Wenn so eine Anpassung stattfindet, findet eine Wissensrevision statt. Diese Revision soll für Multi-Shot Solver [Ge15], eine Form der reaktiven Antwortmengenprogrammierung, genauer untersucht werden. Anders als Arbeiten die sich mit Wissensrevision in der Antwortmengenprogrammierung allgemein beschäftigen, wird in dieser Arbeit besonders die Wissensrevision in dem Multi-Shot Ansatz zur reaktiven Antwortmengenprogrammierung untersucht. Nachdem die notwendigen Grundlagen zur reaktiven Antwortmengenprogrammierung (Kapitel 2) und zur Wissensrevision (Kapitel 3) vorgestellt wurden, werden wir dazu zunächst die AGM-Kriterien [AGM85], die für die Untersuchung der Revision auf Mengen logischer Aussagen entworfen wurden, an die reaktive Antwortmengenprogrammierung anpassen (Kapitel 4). Anschließend untersuchen wir die Wissensrevision bei der reaktiven Antwortmegenprogrammierung mit den angepassten AGM-Kriterien und den Basisrevisionskriterien aus [KK12] (Kapitel 5).

2 Reaktive Antwortmengenprogrammierung

Die *Antwortmengenprogrammierung* [BK14, Kap. 9], englisch *answer set programming* oder kurz *ASP*, ist eine Form der deklarativen Programmierung und basiert auf einer Semantik sogenannter stabiler Modelle (Antwortmengen). Als Grundlage der Antwortmengenprogrammierung dienen *erweiterte logische Programme* [BK14, Kap. 9.5].

Definition 1 (Erweitertes logisches Programm)**.** *Ein* erweitertes logisches Programm $\mathcal{P}$ *ist eine endliche Menge von* Regeln *der Form*

$$r: \quad H \leftarrow P_1, \ldots, P_n, not\ N_1, \ldots, not\ N_m.$$

Dabei sind $H, P_1, \ldots, P_n$ *und* $N_1, \ldots, N_m$ Literale*, d.h. prädikatenlogische Atome aus der Herbrandbasis* $\mathcal{H}(\mathcal{P})$ *oder deren Negation. Mit* $head(r) := \{H\}$ *bezeichnet man den* Kopf *der Regel,* $P_1, \ldots, P_n$ *heißen* positive Rumpfliterale *und* $N_1, \ldots, N_m$ negative Rumpfliterale*. Die Menge aller Rumpfliterale ist* $body(r) := \{P_1, \ldots, P_n, not\ N_1, \ldots, not\ N_m\}$*. Regeln ohne Rumpfliterale heißen* Fakten *und werden ohne* $\leftarrow$ *notiert. Es dürfen auch Regeln mit leerem Kopf vorkommen, diese heißen* Constraints*.*

Eine Regel eines erweiterten logischen Programms ist anschaulich so zu verstehen, dass der Kopf einer Regel erfüllt sein muss, wenn alle positiven Rumpfliterale erfüllt sind und für alle negativen Rumpfliterale nicht sicher ist, dass sie erfüllt sind. Der Rumpf eines Constraints darf nicht erfüllt sein. Um die Semantik formal zu erklären, benötigen wir zunächst den Begriff der *Gelfond-Lifschitz-Reduktion* [BK14, Kap. 9.5]. Sie entfernt sämtliche *Default-Negationen* (*not*-negierte Literale) aus einem erweiterten logischen Programm und reduziert letzteres somit auf ein klassisches logisches Programm.

Definition 2 (Gelfond-Lifschitz-Reduktion)**.** *Sei* $\mathcal{P}$ *ein erweitertes logisches Programm und* S *ein* Zustand*, d.h. eine endliche Menge von Literalen, in der es keine zwei Literale* $A, B \in S$

mit $A = \neg B$ gibt. Die Gelfond-Lifschitz-Reduktion *bildet $\mathcal{P}$ und S auf das folgende* Redukt *ab:*

$$\mathcal{P}^S = \{H \leftarrow P_1, \ldots, P_n. \mid H \leftarrow P_1, \ldots, P_n, not\ N_1, \ldots, not\ N_m. \in \mathcal{P} \\ und\ S \cap \{N_1, \ldots, N_m\} = \emptyset\} \ .$$

Die Reduktion entfernt alle Regeln aus $\mathcal{P}$, bei denen ein negatives Rumpfliteral auch in S vorkommt. Bei den übrigen Regeln werden die negativen Rumpfliterale entfernt. Übrig bleibt ein Programm ohne Default-Negation. Eine *Antwortmenge* [BK14, Kap. 9.7] kann nun als Fixpunkt unter Anwendung des *Cl*-Operators auf das entsprechende Redukt definiert werden. Der *Cl*-Operator bildet dabei ein logisches Programm auf sein minimalen geschlossenen Zustand, d.h. die kleinste Menge von Literalen, die jede Regel erfüllt, ab [BK14, Kap. 9.6].

Definition 3 (Antwortmenge)**.** *Es sei $\mathcal{P}$ ein erweitertes logisches Programm und S ein Zustand. S ist eine* Antwortmenge *von $\mathcal{P}$, wenn $S = Cl(\mathcal{P}^S)$ gilt.*

Anders als bei klassisch logischen Programmen gibt es bei erweiterten logischen Programmen nicht immer eine eindeutige Antwortmenge.

Beispiel 1. *Seien $\mathcal{P}_1 = \{P \leftarrow not\ Q.,\ Q \leftarrow not\ P.\}$ und $\mathcal{P}_2 = \{P \leftarrow not\ Q.,\ Q \leftarrow P.\}$ erweiterte logische Programme. Sowohl $S_a = \{P\}$ als auch $S_b = \{Q\}$ sind Antwortmengen für das Programm $\mathcal{P}_1$: $Cl(\mathcal{P}_1^{S_a}) = Cl(\{P.\}) = \{P\} = S_a$ und $Cl(\mathcal{P}_1^{S_b}) = S_b$. Das Programm $\mathcal{P}_2$ hingegen hat keine Antwortmenge.*

Regeln, die Literale mit Variablen beinhalten, sind als Schemata zu verstehen. Vor dem Bestimmen von Antwortmengen werden diese Regeln *grundiert*, d.h. die Variablen werden durch die zur Verfügung stehenden Konstanten ersetzt. Ist beispielsweise die Konstantenmenge $\{1, 2\}$ gegeben, so wird die Regel $a(X) \leftarrow b(X), c(X).$ durch die beiden Regeln $a(1) \leftarrow b(1), c(1).$ und $a(2) \leftarrow b(2), c(2).$ ersetzt. Ein Programm, das diesen Vorverarbeitungsschritt durchführt, heißt *Grounder*. Der Grounder vereinfacht dabei das entstehende Programm direkt. Käme $b(2)$ beispielsweise nicht im Kopf einer Regel des Programms vor, würde er die Regel $a(2) \leftarrow b(2), c(2).$ nicht erzeugen.

Mittels Kardinalitätsschranken lassen sich Regeln um die Möglichkeit erweitern, mehrere Literale im Kopf zu beinhalten. Eine solche Regel hat dann die Form $k\{H_1, ..., H_j\}l \leftarrow P_1, ..., P_n, not\ N_1, ..., not\ N_m.$ mit natürlichen Zahlen k und l. Wenn der Rumpf der Regel erfüllt ist, kann die Belegung der Literale $H_1, ..., H_j$ im Kopf frei gewählt werden, solange mindestens k und höchstens l der Literale erfüllt sind. Dabei ist es auch möglich, eine oder beide Grenzen wegzulassen. Fehlt die obere Grenze, können beliebig viele Literale im Kopf der Regel wahr sein. Fehlt die untere Grenze, dürfen beliebig wenig Literale erfüllt sein. Insbesondere ist es möglich, dass alle Literale mit „Falsch“ belegt werden. Die Regel $\{H\}.$ bedeutet also, dass H möglicherweise erfüllt ist.

Für manche Anwendungen ist es notwendig, das logische Programm oft zu ändern, beispielsweise wenn Fakten eine sich verändernde Umgebung darstellen. Für „normale“

Antwortmengenprogramme bedeutet eine Änderung, das Programm anzupassen und erneut dem Grounder und dem Solver[2] zu übergeben. *Reaktive Antwortmengenprogrammierung* gestaltet solche Änderungen einfacher und effizienter.

Ein Ansatz zur reaktiven Antwortmengenprogrammierung ist die Multi-Shot Methodik [Ge15]. Dieser Ansatz wurd auch in Clingo[3], einem der wichtigsten Programme zum Lösen von Antwortmengenprogrammen, umgesetzt.

Die Multi-Shot Methodik erlaubt es, die Belegung einiger Literale zu ändern, auch nachdem das Program bereits grundiert wurde. Diese Literale müssen vor dem Grundieren gekennzeichnet werden.

Definition 4 (erweiterbares logisches Programm)**.** *Ein* erweiterbares logisches Programm $\mathcal{P}$ *darf neben Regeln zusätzlich* externe Deklarationen *der Form*

$$\texttt{\#external}\ a\ \texttt{:}\ B.$$

mit einem Atom a und einem Regelrumpf B enthalten. Man bezeichnet a als externe Variable.

Eine externe Variable, die nicht im Kopf einer Regel des erweiterbaren logischen Programms vorkommt, wird vor dem Lösen des Programms mit „Wahr“, „Falsch“ oder „Unbekannt“ belegt. Ohne entsprechende Kennzeichnung würde der Grounder annehmen, dass a falsch ist, weil es nicht im Kopf einer Regel vorkommt und das Programm entsprechend vereinfachen.

Definition 5 (partielle Belegung)**.** *Eine* partielle Belegung *einer Menge A von grundierten Atomen ist eine Funktion* $i : A \to \{t, f, u\}$ *die jedem Atom einen der Werte „Wahr“, „Falsch“ oder „Unbekannt“ (bzw. t, f oder u) zuordnet. Wir schreiben* $A^t = \{a \in A \mid i(a) = t\}$ *und analog* $A^f = \{a \in A \mid i(a) = f\}$ *und* $A^u = \{a \in A \mid i(a) = u\}$*. Üblicherweise gibt man eine partielle Belegung durch* $\langle A^t, A^f \rangle$ *oder* $\langle A^t, A^u \rangle$ *an.*

Der Zustand eines Programmes wird in Clingo während der Ausführung durch ein Tripel $(\mathcal{Q}, \mathcal{P}, I)$ repräsentiert. Dabei ist $\mathcal{Q}$ ein (noch nicht grundiertes) logisches Programm. $\mathcal{Q}$ dient lediglich dazu, Regeln zu sammeln, die zum Programm hinzugefügt werden sollen, bevor sie grundiert werden. $\mathcal{P}$ ist ein grundiertes logisches Programm. $\mathcal{P}$ enthält die Regeln, die das Verhalten des Programms bestimmen. Wenn die Regeln in $\mathcal{Q}$ grundiert wurden, werden sie zu $\mathcal{P}$ hinzugefügt. I schließlich ist die Menge der Input-Atome, d.h. der externen Variablen von $\mathcal{P}$, die nicht im Kopf einer Regel vorkommen, zusammen mit einer partielle Belegung $\langle I^t, I^u \rangle$ dieser Atome. Die Kombination von $\mathcal{P}$ mit der Belegung $\langle I^t, I^u \rangle$, die beim Lösen dem Solver übergeben wird, ist $P_{\langle I^t, I^u \rangle} := P \cup \{a. \mid a \in I^t\} \cup \{\{a\}. \mid a \in I^u\}$.

Um den Zustand eines Programms zu beeinflussen, stehen verschiedene Operationen zur Verfügung. Hier sind vor allem die Operationen add, ground und assignExternal interessant. In [Ge15] sind ausserdem die Operationen create und releaseExternal, sowie solve zum Lösen von Programmen beschrieben.

[2] Der Solver ist die Software, die ein Antwortmengenprogramm löst indem es passende Antwortmengen bestimmt.
[3] Clingo ist Teil der Softwaresammlung *Potassco* die an der Universität Potsdam entwickelt wird (potassco.org)

add($\mathcal{R}$) $(\mathcal{Q}, \mathcal{P}, I) \mapsto (\mathcal{Q} \cup \mathcal{R}, \mathcal{P}, I)$ für ein (nicht grundiertes) Programm $\mathcal{R}$
Fügt nicht grundierte Regeln zu dem Programm hinzu.

ground() $(\mathcal{Q}, \mathcal{P}_1, I_1) \mapsto (\emptyset, \mathcal{P}_2, I_2)$ mit

- $D = \{a \leftarrow B, \varepsilon. \mid \texttt{\#external } a \texttt{ : } B. \in \mathcal{Q}\}$
  $\mathcal{P} = \{r \in grd(\mathcal{Q} \cup D \cup \{\{\varepsilon\}.\}) \setminus \{\{\varepsilon\}.\} \mid \varepsilon \notin body(r)\}$
  $E = \{head(r) \mid r \in grd(\mathcal{Q} \cup D \cup \{\{\varepsilon\}.\}),\ \varepsilon \in body(r)\}$
  $\mathcal{P}$ enthält die grundierten Regeln aus $\mathcal{Q}$ und E die in $\mathcal{Q}$ deklarierten externen Variablen. ε ist ein spezielles Atom, das Regeln aus externen Deklarationen markiert und in $\mathcal{Q}$ nicht vorkommt. Weitere Details hierzu finden sich in [Ge15].
- $\mathcal{P}_2 = \mathcal{P}_1 \cup \mathcal{P}$
- $I_2 = (I_1 \cup E) \setminus head(\mathcal{P}_2)$
- $I_2^t = I_2 \cap I_1^t, \qquad I_2^u = I_2 \cap I_1^u, \qquad I_2^f = I_2 \setminus I_2^t \setminus I_2^u.$

Grundiert das Programm in $\mathcal{Q}$ bzgl. $\mathcal{P}_1$ und I_1, fügt es zu $\mathcal{P}$ hinzu und passt die partielle Belegung der Input-Variablen I an.
Bei der Grundierung von $\mathcal{Q}$ müssen $\mathcal{P}_1$ und I_1 einfließen, da sie zur Herbrandbasis (d.h. die verfügbaren Konstanten) beitragen. Zu I werden die neuen externen Variablen aus $\mathcal{Q}$ hinzugefügt. Die externen Variablen, die in einem Regelkopf von $\mathcal{Q}$ vorkommen, werden aus den Inputatomen entfernt.

assignExternal(a, x) $(\mathcal{Q}, \mathcal{P}, I_1) \mapsto (\mathcal{Q}, \mathcal{P}, I_2)$ mit

- $I_2 = I_1$
- $I_2^t = I_1^t \cup \{a\}$ wenn $x = t$ und $a \in I_1$ ist
  $I_2^t = I_1^t$ wenn $x = t$ und $a \notin I_1$ ist
  $I_2^t = I_1^t \setminus \{a\}$ wenn $x \in \{f, u\}$
- $I_2^u = I_1^u \cup \{a\}$ wenn $x = u$ und $a \in I_1$ ist
  $I_2^u = I_1^u$ wenn $x = u$ und $a \notin I_1$ ist
  $I_2^u = I_1^u \setminus \{a\}$ wenn $x \in \{t, f\}$
- $I_2^f = I_2 \setminus I_2^t \setminus I_2^u$.

Belegt das Atom a mit x, wobei x „wahr“ (t), „falsch“ (f) oder „unbekannt“ (u) sein kann. Für $x = t$ oder $x = f$ kommt a in jeder bzw. keiner Antwortmenge vor. Für $x = u$ ist nicht festgelegt, ob a in den Antwortmengen des Programms vorkommen muss oder nicht.

3 Wissensrevision

Es kommt vor, dass Software-Agenten ihr Wissen anpassen müssen. Die AGM-Theorie [AGM85] nimmt an, dass das Wissen eines Agenten eine Menge von (logischen) Aussagen

ist. Ist diese Menge über dem Cn-Operator[4] abgeschlossen, bezeichnet man sie auch als Wissensmenge.

Auf einer Menge von Aussagen sind die Operationen Expansion (+), Kontraktion (−) und Revision (∗) möglich. Bei der Expansion werden neue Informationen zu dem vorhandenen Wissen hinzugefügt. Bei der Kontraktion werden bestimmte Aussagen aus dem vorhanden Wissen entfernt, sollen also vergessen werden. Bei der Revision sollen neue Informationen hinzugefügt werden. Dabei soll jedoch das vorhandene Wissen so angepasst werden, dass das Wissen nach der Operation wieder konsistent ist, auch wenn die neuen Informationen dem alten Wissen wiedersprechen.

Um verschiedene Verfahren für diese Operationen zu bewerten, werden die sogenannten AGM-Kriterien verwendet. Diese sind wünschenswerte Eigenschaften von Verfahren für die beschriebenen Operationen.

Definition 6 (AGM-Revisionskriterien)**.** *Es sei A eine Menge von logischen Aussagen und x eine neue Aussage. Die AGM-Kriterien für die Revision nach [AGM85] lauten:*

(*1) $A * x$ *ist eine Wissensmenge.*

(*2) $x \in A * x$

(*3a) $A * x \subseteq Cn(A \cup \{x\})$

(*3b) *Wenn* $\neg x \notin Cn(A)$ *gilt, dann ist* $Cn(A \cup \{x\}) \subseteq A * x$.

(*4) *Wenn x erfüllbar ist, dann ist* $A * x$ *konsistent.*

(*5) *Wenn* $Cn(x) = Cn(y)$ *gilt, dann ist* $A * x = A * y$.

(*6) *Wenn A eine Wissensmenge ist, dann ist* $(A * x) \cap A = A - \neg x$.

(*7) $A * (x \wedge y) \subseteq Cn((A * x) \cup \{y\})$

(*8) *Wenn* $\neg y \notin A * x$ *gilt, dann ist* $Cn((A * x) \cup \{y\}) \subseteq A * (x \wedge y)$.

*Die Kriterien (*3a) und (*3b) sind in der Originalarbeit zusammengefasst, werden in einigen Quellen (z.B. [Ke17]) jedoch als einzelne Kriterien aufgeführt.*

Das Kriterium AGM (*2) beispielsweise fordert, dass das neue Wissen x in dem Wissen nach der Revision enthalten ist; AGM (*5) fordert, dass semantisch äqivalente Informationen den gleichen Einfluss auf die Menge der Aussagen nach der Revision haben.

[4] Der Cn-Operator bildet eine Menge M von logischen Aussagen auf die Menge $Cn(M) := \{a \mid M \models a\}$ ab.

Man sieht, dass die AGM-Kriterien nicht für Antwortmengenprogrammierung vorgesehen sind. Ein logisches Programm ist eine Menge von Regeln und nicht von Aussagen. In Kapitel 4 werden wir die AGM-Revisionskriterien an reaktive Antwortmengenprogrammierung anpassen. Eine andere Alternative zu den AGM-Revisionskriterien sind die Basisrevisionskriterien für Antwortmengenprogramme, die in [KK12] vorgestellt wurden.

Definition 7 (Basisrevisionskriterien)**.** *Im Folgenden sei P das vorhandene logische Programm und Q eine Menge von neuen Regeln, die hinzugefügt werden sollen. Dann sind die Basisrevisionskriterien für Antwortmengenprogrammierung:*

Success $Q \subseteq P * Q$

Inclusion $P * Q \subseteq P \cup Q$

Vacuity *Wenn $P \cup Q$ konsistent*[5] *ist, dann gilt $P \cup Q \subseteq P * Q$.*

Consistency *Wenn Q konsistent ist, dann ist $P * Q$ konsistent.*

NM-Consistency *Wenn es ein konsistentes Programm X gibt, so dass $Q \subseteq X \subseteq P \cup Q$ gilt, dann ist $P * Q$ konsistent.*

Relevance *Wenn es eine Regel $r \in (P \cup Q) \setminus (P * Q)$ gibt, dann existiert ein Programm H mit $P * Q \subseteq H \subseteq P \cup Q$, so dass H konsistent ist, aber $H \cup \{r\}$ nicht.*

Fullness *Wenn es eine Regel $r \in (P \cup Q) \setminus (P * Q)$ gibt, dann ist $P * Q$ konsistent, $(P * Q) \cup \{r\}$ aber nicht.*

Uniformity *Wenn für alle $P' \subseteq P$ gilt, dass $P' \cup Q$ inkonsistent ist genau dann wenn $P' \cup R$ inkonsistent ist, dann ist $P \cap (P * Q) = P \cap (P * R)$.*

Dabei gilt, dass Fullness strenger ist als Relevance und Relevance strenger ist als Vacuity: Fullness $\Rightarrow$ Relevance $\Rightarrow$ Vacuity. Außerdem folgt Consistency aus NM-Consistency. Die unterschiedlich starken Versionen der Kriterien erlauben es untersuchten Operationen genauer einzuordnen.

4 Anpassung der Revisionskriterien für die reaktive Antwortmengenprogrammierung

Die AGM-Kriterien sind zur Beurteilung der Wissensrevision auf Mengen von Aussagen entworfen. Deshalb lassen sich die meisten Kriterien nicht auf die Operationen der reaktiven

[5] Ein (erweitertes) logisches Programm $\mathcal{P}$ heißt genau dann *konsistent* oder *erfüllbar*, wenn $\mathcal{P}$ mindestens eine Antwortmenge hat.

Antwortmengenprogrammierung anwenden. Im Folgenden versuchen wir die Revisionskriterien so anzupassen, dass sie auch auf die Operationen ground und assignExternal der Multishotsolver anwendbar sind. Die Herausforderung ist dabei, die den Kriterien zugrundeliegende Struktur auf die reaktive Antwortmengenprogrammierung zu übertragen, obwohl viele Konzepte wie beispielsweise die Abgeschlossenheit von Aussagenmengen bei Antwortmengenprogrammen nicht anwendbar sind.

Dafür sei $T_1 = (Q_1, \mathcal{P}_1, I_1)$ das Clingo-Tripel vor und $(Q_2, \mathcal{P}_2, I_2)$ das Tripel nach Anwendung der jeweiligen Operation. Q sei ein logisches Programm, welches das neu hinzuzufügende Wissen repräsentiert. Während bei dem Kriterium (*1') die Tripel-Struktur berücksichtigt werden kann, werden wir bei den meisten Kriterien statt der Clingo-Tripel die Mengen $\mathcal{P}_{1\langle I_1^t, I_1^u\rangle}$ und $\mathcal{P}_{2\langle I_2^t, I_2^u\rangle}$ betrachten:

$$T_1 = (Q_1, \mathcal{P}_1, I_1) \xrightarrow{\quad *Q \quad} T_2 = (Q_2, \mathcal{P}_2, I_2)$$

$$\mathcal{P}_{1\langle I_1^t, I_1^u\rangle} \dashrightarrow \mathcal{P}_{2\langle I_2^t, I_2^u\rangle}$$

Untersuchte Revision

Im Folgenden sind beispielhaft die Überlegungen zu den Anpassungen der AGM-Revisionskriterien (*3b), (*5) und (*6) dargestellt.

AGM-Kriterium (*3b): Wenn $\neg x \notin Cn(A)$ gilt, dann ist $Cn(A \cup \{x\}) \subseteq A * x$.
Die Vorraussetzung $\neg x \notin Cn(A)$ lässt sich als „x widerspricht A nicht" interpretieren. Für Antwortmengenprogramme P, Q lässt sich dass durch „$P \cup Q$ ist konsistent" ausdrücken. Die Folgerung $Cn(A \cup \{x\}) \subseteq A * x$ drückt aus, das alles, was sich aus aus $A \cup \{x\}$ folgern lässt, auch in dem revidierten Wissen $A * x$ enthalten ist. Der Cn-Operator lässt sich nicht auf Antwortmengenprogramme anweden, aber wir können fordern, dass alle alten und neuen Regeln bei der Revision erhalten bleiben: $P \cup Q \subseteq P * Q$.

Schließlich ersetzen wir die Mengen A und $A * x$, wie beschrieben, durch $\mathcal{P}_{1\langle I_1^t, I_1^u\rangle}$ vor und $\mathcal{P}_{2\langle I_2^t, I_2^u\rangle}$ nach der Operation. Das neue Kriterium lautet:
AGM (*3b'): Wenn $\mathcal{P}_{1\langle I_1^t, I_1^u\rangle} \cup Q$ konsistent ist, dann ist $\mathcal{P}_{1\langle I_1^t, I_1^u\rangle} \cup Q \subseteq \mathcal{P}_{2\langle I_2^t, I_2^u\rangle}$.

AGM-Kriterium (*5): Wenn $Cn(x) = Cn(y)$ gilt, dann ist $A * x = A * y$.
$Cn(x) = Cn(y)$ bedeutet, dass x und y äquivalent sind. Für Antwortmengenprogramme P, Q gibt es verschiedene Arten von Äquivalenz. Eine der einfachsten ist $C_{AS}(P) = C_{AS}(Q)$ mit $C_{AS}(P) := \{L \mid L \text{ liegt in allen Antwortmengen von } P\}$.

Sei $T_2 = (Q_2, \mathcal{P}_2, I_2)$ das Ergebnis der Revision mit Q und $T_3 = (Q_3, \mathcal{P}_3, I_3)$ das Ergebnis der Revision von T_1 mit R. Dann ist das angepasste Kriterium:
AGM (*5'): Wenn $C_{AS}(Q) = C_{AS}(R)$ gilt, dann ist $\mathcal{P}_{2\langle I_2^t, I_2^u\rangle} = \mathcal{P}_{3\langle I_3^t, I_3^u\rangle}$ mit $T_3 = T_1 * Q$.

AGM-Kriterium (*6): Wenn A eine Wissensmenge ist, dann ist $(A * x) \cap A = A - \neg x$. Dieses Kriterium fordert, dass von den Aussagen in der alten Wissensmenge bei der Revision nur die entfernt werden, die für das Vergessen von $\neg x$ entfernt werden müssen. Es gibt aber keine Negation und keine Kontraktion für allgemeine logische Programme. Dieses Kriterium können wir zur Untersuchung der Antwortmengenprogramme also nicht sinvoll anpassen. Die AGM (*6) zugrundeliegende Idee, dass bei der Revision keine Aussagen unnötigerweise entfernt werden, wird am ehesten in den Basisrevisionskriterien Fullness und Relevance ausgedrückt.

Die anderen AGM-Revisionskriterien lassen sich in ähnlicher Weise anpassen. Wir haben uns für die folgenden angepassten Kriterien entschieden:

(*1') Die Revisionsoperation gibt ein Clingo-Tripel zurück.

(*2') $Q \subseteq \mathcal{P}_{2\langle I_2^t, I_2^u\rangle}$

(*3a') $\mathcal{P}_{2\langle I_2^t, I_2^u\rangle} \subseteq \mathcal{P}_{1\langle I_1^t, I_1^u\rangle} \cup Q$

(*3b') Wenn $\mathcal{P}_{1\langle I_1^t, I_1^u\rangle} \cup Q$ konsistent ist, dann ist $\mathcal{P}_{1\langle I_1^t, I_1^u\rangle} \cup Q \subseteq \mathcal{P}_{2\langle I_2^t, I_2^u\rangle}$.

(*4') Wenn Q konsistent ist, dann ist $\mathcal{P}_{2\langle I_2^t, I_2^u\rangle}$ konsistent.

(*5') Wenn $C_{AS}(Q) = C_{AS}(R)$ gilt, dann ist $\mathcal{P}_{2\langle I_2^t, I_2^u\rangle} = \mathcal{P}_{3\langle I_3^t, I_3^u\rangle}$ mit $T_3 = T_1 * Q$.

Bei den Kriterien (*7) und (*8) hängen die Anpassungen von der Operation ab:

(*7') $\mathcal{P}_{5\langle I_5^t, I_5^u\rangle} \subseteq \mathcal{P}_{2\langle I_2^t, I_2^u\rangle} \cup R$ mit $T_5 = (T_1 * Q) * R$ bei der Untersuchung von *assignExternal*

(*7'') $\mathcal{P}_{4\langle I_4^t, I_4^u\rangle} \subseteq \mathcal{P}_{2\langle I_2^t, I_2^u\rangle} \cup R$ mit $T_4 = T_1 * (Q \cup R)$ bei der Untersuchung von *ground*

(*8') Wenn $\mathcal{P}_{2\langle I_2^t, I_2^u\rangle} \cup R$ konsistent ist, dann gilt $\mathcal{P}_{2\langle I_2^t, I_2^u\rangle} \cup R \subseteq \mathcal{P}_{5\langle I_5^t, I_5^u\rangle}$ mit $T_5 = (T_1 * Q) * R$ für die Untersuchung von *assignExternal*

(*8'') Wenn $(\mathcal{P}_{2\langle I_2^t, I_2^u\rangle} \cup R$ konsistent ist, dann gilt $\mathcal{P}_{2\langle I_2^t, I_2^u\rangle} \cup R \subseteq \mathcal{P}_{4\langle I_4^t, I_4^u\rangle}$ mit $T_4 = T_1 * (P \cup Q)$ für die Untersuchung von *ground*.

Die Basisrevisionskriterien sind bereits an die Verwendung mit Antwortmengenprogrammen angepasst. Sie sind jedoch nicht für die Untersuchung von Clingo-Tripeln vorgesehen. Seien $T_1 = (Q_1, \mathcal{P}_1, I_1)$ und $T_2 = (Q_2, \mathcal{P}_2, I_2)$ die Clingo-Tripel vor und nach der Operation. Es werden daher bei den Untersuchungen mit den Basisrevisionskriterien statt der Clingo-Tripel T_1 und T_2 die Mengen $P := \mathcal{P}_{1\langle I_1^t, I_1^u\rangle}$ und $P * Q := \mathcal{P}_{2\langle I_2^t, I_2^u\rangle}$ betrachtet.

Damit sind einige Basisrevisionskriterien äquivalent zu angepassten AGM-Kriterien. Success ist äquivalent zu AGM-Kriterium (*2'), Inclusion ist äquivalent zu AGM-Kriterium (*3a'), Vacuity ist äquivalent zu AGM-Kriterium (*3b') und Consistency ist äquivalent zu AGM-Kriterium (*4').

5 Formale Untersuchung der reaktiven Antwortmengenprogrammierung auf Erfüllung der Revisionskriterien

Untersuchen wir zunächst die Operation *assignExternal(v, t)* mit den angepassten AGM-Revisionskriterien aus Kapitel 4 und den Basisrevisionskriterien. Diese Operation belegt die externen Variable v mit „wahr“. Das neu dazugewonnene Wissen ist also $Q := \{v.\}$.

Satz 1. *Die Operation* assignExternal(v, t) *erfüllt die angepassten AGM-Kriterien (*1'), (*2'), (*3a'), (*5') und (*7'). Die Kriterien (*3b'), (*4') und (*8') sind nicht erfüllt.*

Die Operation erfüllt die Basisrevisionskriterien Success und Inclusion. Die Kriterien Vacuity, Relevance, Fullness, (NM-)Consistency und Uniformity sind nicht erfüllt.

Beweis. Für diese Operation gilt $\mathcal{P}_{2\langle I_2^t, I_2^u\rangle} = \mathcal{P}_{1\langle I_1^t, I_1^u\rangle} \cup \{v.\} \setminus \{\{v\}.\}$.

Die Operation *assignExternal(v, t)* gibt ein Clingo-Tripel zurück. AGM (*1') ist also erfüllt.
Nach der Operation gilt $I_2(v) = t$. Damit ist $Q = \{v.\} \subseteq \mathcal{P}_2 \cup \{a. \mid a \in I_2^t\} \cup \{\{a\}. \mid a \in I_2^u\} = \mathcal{P}_{2\langle I_2^t, I_2^u\rangle}$. Also ist AGM (*2') erfüllt.
Wenn $I_1(v) = t$ oder $I_1(v) = f$ gilt, dann ist $\mathcal{P}_{2\langle I_2^t, I_2^u\rangle} = \mathcal{P}_{1\langle I_1^t, I_1^u\rangle} \cup \{v.\} = \mathcal{P}_{1\langle I_1^t, I_1^u\rangle} \cup Q$ und das Kriterium (*3a') ist erfüllt. Wenn aber $I_1(v) = u$ gilt, dann ist $\mathcal{P}_{2\langle I_2^t, I_2^u\rangle} = \mathcal{P}_{1\langle I_1^t, I_1^u\rangle} \cup \{v.\} \setminus \{\{v\}.\} \subseteq \mathcal{P}_{1\langle I_1^t, I_1^u\rangle} \cup Q$ und das Kriterium (*3a') ist ebenfalls erfüllt.
AGM (*3b') ist nicht erfüllt. Wenn vor der Operation $I_1(v) = u$ gilt, ist $\mathcal{P}_{2\langle I_2^t, I_2^u\rangle} = \mathcal{P}_{1\langle I_1^t, I_1^u\rangle} \cup Q \setminus \{\{v\}.\} \subsetneq \mathcal{P}_{1\langle I_1^t, I_1^u\rangle} \cup Q$.
Ein Gegenbeispiel für AGM (*4') ist:

Beispiel 2. *Sei* $Q_1 = \emptyset$ *und* $\mathcal{P}_1 = \{\leftarrow v.\}$ *und* $I_1 = \{v \mapsto f\}$. *Die Antwortmenge von* $\mathcal{P}_{1\langle I_1^t, I_1^u\rangle}$ *ist* $\emptyset$. *Nach der Operation* assignExternal(v, t) *erhalten wir das Tripel* $(Q_2, \mathcal{P}_2, I_2)$ *mit* $I_2(v) = t$. *Das Programm* $\mathcal{P}_{2\langle I_2^t, I_2^u\rangle} = \{\leftarrow v., v.\}$ *hat keine Antwortmenge, obwohl* $Q = \{v.\}$ *konsistent ist.*

Für zwei einelementige Programme $Q = \{v.\}$ und $R = \{w.\}$ gilt $C_{AS}(Q) = C_{AS}(R)$ genau dann, wenn $v = w$ gilt. Wenn $v = w$ gilt, dann auch $\mathcal{P}_{2\langle I_2^t, I_2^u\rangle} = \mathcal{P}_{3\langle I_3^t, I_3^u\rangle}$. AGM (*5') ist also erfüllt.
AGM (*7') ist erfüllt: $\mathcal{P}_{5\langle I_5^t, I_5^u\rangle} = \mathcal{P}_5 \cup \{a. \mid a \in I_5^t\} \cup \{\{a\}. \mid a \in I_5^u\}$
$= \mathcal{P}_2 \cup \{a. \mid a \in I_2^t\} \cup \{\{a\}. \mid a \in I_2^u\} \cup \{w.\} \setminus \{\{w\}.\} \subseteq \mathcal{P}_{2\langle I_2^t, I_2^u\rangle} \cup \{w.\}$
AGM (*8') ist nicht erfüllt. Wenn vor der Operation $I_1(w) = u$ war, dann ist $\{w\}. \in \mathcal{P}_{2\langle I_2^t, I_2^u\rangle}$ aber $\{w\}. \notin \mathcal{P}_{5\langle I_5^t, I_5^u\rangle}$.

Die Basisrevisionskriterien Success, Inclusion, Vacuity und Consistency sind jeweils äquivalent zu einem bereits untersuchten angepassten AGM-Kriterium. Die Kriterien Fullness und Relevance sind nicht erfüllt, da schon das schwächere Kriterium Vacuity nicht erfüllt ist. Ebenso kann NM-Consistency nicht erfüllt sein, da Consistency nicht erfüllt ist.

Uniformity ist nicht erfüllt, wie dieses Gegenbeispiel zeigt:

Beispiel 3. *Sei* $\mathcal{P}_1 = \mathcal{Q}_1 = \emptyset$ *und* $I_1 = \{v \mapsto u, w \mapsto u\}$. *Die Anwendung der Operation* assignExternal(v, t) *(also* $Q := \{v.\}$*) führt zu dem Tripel* $(\mathcal{P}_2, \mathcal{Q}_2, I_2)$, *die Anwendung von* assignExternal(w, t) *(also* $R := \{w.\}$*) führe zu* $(\mathcal{P}_3, \mathcal{Q}_3, I_3)$. *Entsprechend sind* $P * Q := \mathcal{P}_{2\langle I_2^t, I_2^u \rangle}$ *und* $P * R := \mathcal{P}_{3.\langle I_3^t, I_3^u \rangle}$.

Es ist $P = \mathcal{P}_{1\langle I_1^t, I_1^u \rangle} = \{\{v\}., \{w\}.\}$. *Damit ist für jede Teilmenge* P' *von* P *sowohl* $P' \cup Q$ *als auch* $P' \cup R$ *konsistent. Trotzdem ist* $P \cap (P * Q) = \{\{v\}., \{w\}.\} \cap \{v., \{w\}.\} = \{\{w\}.\} \neq \{\{v\}.\} = \{\{v\}., \{w\}.\} \cap \{\{v\}., w.\} = P \cap (P * R)$.

□

Diese Untersuchungen lassen sich analog für *ground()* und *assignExternal(v, f)* mit $\mathcal{Q} := \mathcal{P}$ bzw. $\mathcal{Q} := \{\neg v\}$ durchführen. Man erhält die Ergebnisse:

Satz 2. *Die Operation* ground() *erfüllt die angepassten AGM-Kriterien (*1'), (*2'), (*3a') und (*7'). Die Kriterien (*3b'), (*4'), (*5) und (*8') sind nicht erfüllt.*

ground() *erfüllt nur die Basisrevisionskriterien Success und Inclusion. Die Kriterien Vacuity, Relevance, Fullness, (NM-)Consistency und Uniformity sind nicht erfüllt.*

Satz 3. *Die Operation* assignExternal(v, f) *erfüllt die angepassten AGM-Kriterien (*1'), (*3a'), (*5') und (*7'). Die Kriterien (*2'), (*3b'), (*4') und (*8') sind nicht erfüllt.*

assignExternal(v, f) *erfüllt nur das Basisrevisionskriterium Inclusion. Die Kriterien Success, Vacuity, Relevance, Fullness, (NM-)Consistency und Uniformity sind nicht erfüllt.*

Die Beweise zu den Aussagen in Satz 2 und 3 befinden sich in [Ha17]. Die Operation *assignExternal(v, u)* ist eine Kontraktion und wurde daher nicht mit den Revisionskriterien untersucht.

6 Zusammenfassung und Ausblick

In diesem Paper wurde die Wissensrevision in der Multi-Shot Methodik untersucht. Dabei kamen hier angepasste AGM-Revisionskriterien und die Basisrevisionskriterien aus [KK12] zum Einsatz. Eine Übersicht über die Ergebnisse findet sich in Tabelle 1. Viele Kriterien sind nicht erfüllt. Die reaktive Antwortmengenprorammierung ist also kein besonders gute geeignetes Werkzeug zum Wissensmanagement.

Offen bleibt die Untersuchung der Operationen mit den AGM-Kontraktionskriterien. Diese wurde in [Ha17] durchgeführt.

Danksagung. Dieses Paper ist auf Grundlage meiner Bachelorarbeit [Ha17] entstanden, die ich unter der Betreuung von Prof. Gabriele Kern-Isberner und Marco Wilhelm verfasst habe. Ich danke allen, die mich beim Verfassen dieser Bachelorarbeit unterstützt haben.

	assignExternal(v, t)	*assignExternal(v, f)*	*ground()*
(*1')	j	j	j
(*2')	j	n	j
(*3a')	j	j	j
(*3b')	n	n	n
(*4')	n	n	n
(*5')	j	j	n
(*7')	j	j	j
(*8')	n	n	n
Success	j	n	j
Inclusion	j	j	j
Vacuity	n	n	n
Relevance	n	n	n
Fullness	n	n	n
Consistency	n	n	n
NM-Consistency	n	n	n
Uniformity	n	n	n

Tab. 1: Übersicht über die Ergebnisse der Untersuchung

Literatur

[AGM85] Alchourrón, C. E.; Gärdenfors, P.; Makinson, D.: On the logic of theory change: Partial meet contraction and revision functions. The Journal of Symbolic Logic 50/02, S. 510–530, 1985.

[BK14] Beierle, C.; Kern-Isberner, G.: Methoden wissensbasierter Systeme: Grundlagen, Algorithmen, Anwendungen. Springer-Verlag, 2014.

[Ge11] Gebser, M.; Grote, T.; Kaminski, R.; Schaub, T.: Reactive answer set programming. In: International Conference on Logic Programming and Nonmonotonic Reasoning. Springer, S. 54–66, 2011.

[Ge15] Gebser, M.; Kaminski, R.; Obermeier, P.; Schaub, T.: Ricochet robots reloaded: A case-study in multi-shot ASP solving. In: Advances in Knowledge Representation, Logic Programming, and Abstract Argumentation. Springer, S. 17–32, 2015.

[Ha17] Haldimann, J. P.: Reaktive Antwortmengenprogrammierung - formale Eigenschaften und logistische Anwendung, Bachelorarbeit, TU Dortmund, 2017.

[Ke17] Kern-Isberner, G.: Darstellung, Verarbeitung und Erwerb von Wissen, Vorlesungsfolien, WS 16/17, URL: https://ls1-www.cs.tu-dortmund.de/images/courses/ie/ws1617/dvew/folien/, Stand: 05. 09. 2017.

[KK12] Krümpelmann, P.; Kern-Isberner, G.: Belief base change operations for answer set programming. In: European Workshop on Logics in Artificial Intelligence. Springer, S. 294–306, 2012.

Community Detection in Complex Networks using Genetic Algorithms

Simon Lehnerer[1]

Abstract: Detecting the community structure is of great interest when analyzing the topology of a network, however it is not a trivial problem. In this article a genetic algorithm is proposed which finds the community structure of a network based on the maximization of a quality function called modularity. Tests using several sample networks show that it reliably finds the community structure. However it does not resolve sufficiently small communities as intuitively expected due to an effect known as *resolution limit*.

Keywords: complex networks; community detection; genetic algorithm

1 Introduction

Broadly speaking, the *community structure* of a network is the division of the network into groups of nodes called *communities*, which are internally densely connected and loosely connected to nodes from other communities. There is no unique formalized definition of the term *community* (*structure*) [Fo10]. In fact, many different approaches to define and detect communities in a network have been developed in the past [FH16; Fo10; Sc07]. We will give a few examples in the following: A simple method to find communities is to divide the nodes of a network in g groups of predefined size such that the total number of edges connecting different groups in minimized [Po97]. For networks with a known hierarchical structure, e.g. often found in social networks, hierarchical clustering algorithms have been developed. Based on a distance measure between nodes they either start by considering the whole network as one single community and then iteratively divide the community into smaller communities (*top-down*), or they start by considering each node as a single community and iteratively merge them into larger communities (*bottom-up*) [HTF09]. Another popular ansatz is to determine communities using the eigenvalues of the Laplacian matrix of the network [Lu07]. Methods based on statistical inference like the *stochastic block model*, a generative model for random graphs, are also often used to study the community structure of a network [KN11]. Several information theoretic approaches try to uncover the community structure by investigating the trace of random walkers on a network and utilize the fact that the walkers are "trapped" inside communities [RB08]. Because the community problem is ill-defined there is no superior approach. It rather depends on the specific network that is

[1] ETH Zürich, Rämistrasse 101, 8092 Zürich, Switzerland, lehsimon@ethz.ch

to be studied and the notion of *community* that is used by a specific method. For certain networks some approaches might be more useful than others [FH16].

In this work we will focus on a quality function called *modularity* [FC12]. The modularity of a community partition is a measure that is the higher the "better" a partition indicates the community structure of the network. Thus the community structure is given by the community partition with maximal modularity.

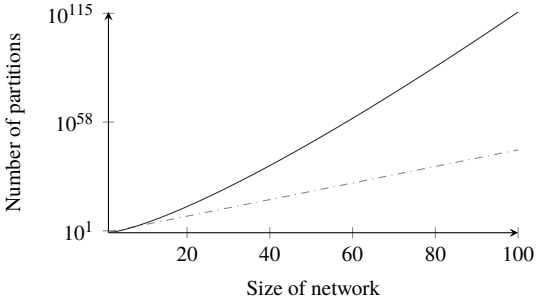


Fig. 1: Number of possible partitions (blue) compared with exponential curve (red). For a network consisting of twelve nodes there already exist more than one million possible community partitions.

There is no method known that can determine this community partition in a simple and straightforward way. That is why modularity based approaches are in fact optimization approaches that try to maximize modularity [FH16]. The number of possible community partitions of a network with n nodes is given by the Bell number B_n, which has factorial growth (see fig. 1). A "brute-force" approach to find the community structure by comparing all possible partitions clearly becomes impractical for larger networks. In addition to that, the modularity function features a high degeneracy which makes it difficult to find its global maximum [Ba16]. For that reason we need an algorithm that can find the community partition with maximal modularity both efficiently and effectively. For this purpose a genetic algorithm (see fig. 2) is proposed in the following.

2 The algorithm

The algorithm [Le18] described here is aimed at finding the community structure of an undirected, unweighted network. A *chromosome* represents a possible community partition of the network. Technically it is a list where the i-th entry contains the community label c_i of node i (see fig. 3). Thus two nodes i and j are in the same community iff $c_i = c_j$. The chromosomes of the initial population are generated randomly.

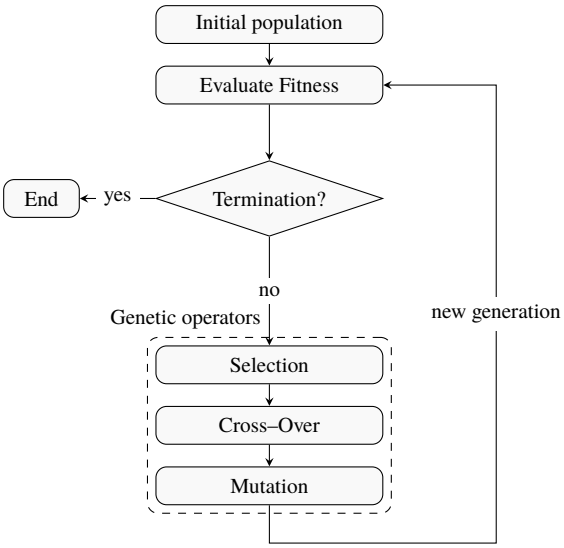


Fig. 2: Schematic of a genetic algorithm
Genetic algorithms, inspired by processes of evolution, are stochastic optimization methods used to find an optimal solution in a large solution space. They use a *population* of *individuals* where each *chromosome* of an individual represents a possible solution. A *fitness* value reflects the quality of the solution of an individual and is used to stochastically perform *selection*, *crossover* and *mutation* operations to form a new generation of individuals (also called *children*). Ideally, the quality of the solutions improves with every new generation. The process is repeated until a reasonable solution is found.

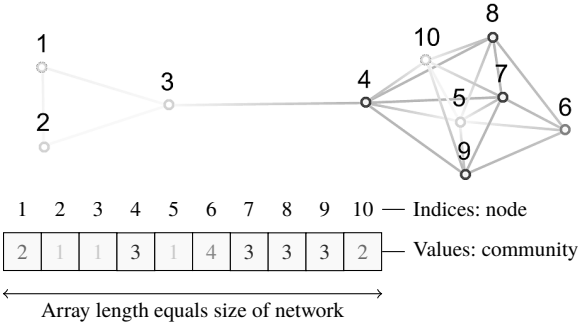


Fig. 3: Example of a chromosome for a network of size 10. It represents a possible community partition of the network (the communities are color-coded so that two nodes are in the same community iff their color is the same).

The fitness is calculated from the modularity of the community partition, which is a global measure of the network and given by

$$Q = \frac{1}{2m} \sum_{i,j} \left(A_{ij} - \frac{d_i d_j}{2m} \right) \delta(c_i, c_j) \quad \in [-0.5, 1]$$

Here m is the number of total links, A_{ij} is the adjacency matrix (which is 1 if there is a connection between nodes i and j, otherwise 0), d_k is the degree of node k, $c_k = l$ if node k is in community l, and $\delta(c_i, c_j) = 1$ if $c_i = c_j$, otherwise 0 [Ne06]. The formula compares the number of internal links in a community with the expected number of links of a randomized graph and therefore implies a definition of the term *community*: A community is a group of nodes that is more tightly connected than expected at random [FC12].

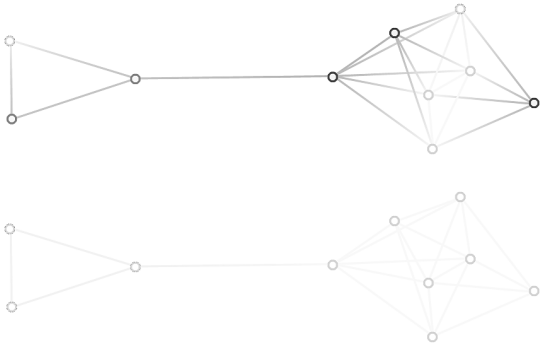


Fig. 4: The figure shows two possible community partitions of a network. While the upper partition has a low modularity of about −0.08, the partition below it has a modularity of about 0.22 which is maximal among all possible partitions and thus shows the community structure of the network.

The algorithm provides two methods of selection. The first, in the following called *Elitism*, only keeps the fittest individual. All children are unaltered copies of this single individual. The second method, in the following called *Mating*, retains a certain fraction of the fittest individuals (*parents*), and performs cross-over operations to create new children (see figure 5 for a detailed description). Both methods are indeed a variant of *elitism*, where only the fittest individuals are preserved. Important to note is that the community partition is a relational property, meaning that the labeling of the communities is ambiguous. Two individuals could actually refer to the same community but use different community labels. This is considered by the algorithm by "translating" the community label when doing cross-over operations.

The mutation process consists of two independent parts and is applied to each child by a certain chance. The first mutation process randomly alters parts of the chromosome (see fig. 6a). In the second mutation process a random sample of nodes will inherit the community that is most dominant in their respective neighborhood (the nodes they are directly connected with).

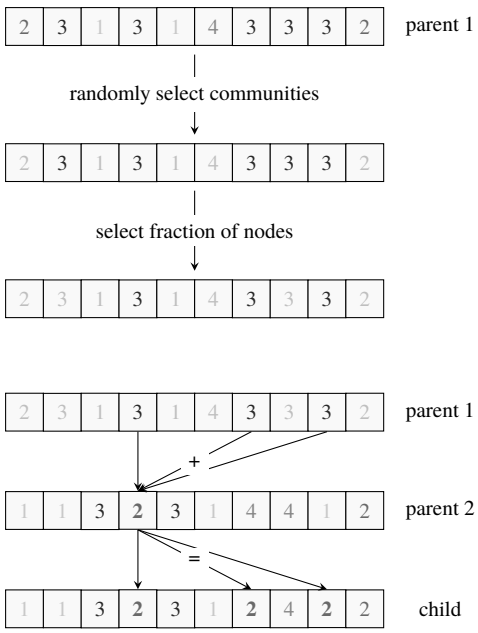


Fig. 5: Cross-over process

The aim of the cross-over process is to combine the chromosomes of two individuals (called *parents*) in order to create a new chromosome (called *child*). This is done in the following way: At first a number of communities of *parent 1* are randomly selected. From the selected communities only a certain fraction of the nodes are then chosen. These nodes are then transferred to *parent 2*, i.e. for those nodes we replace their respective community label of *parent 2* by the respective community label of *parent 1*. Since the community labelling is a relational property, we "translate" the community label of *parent 1*. For each community to be transferred it uses the corresponding community label of the **first node** in *parent 2* as community label (indicated by the arrows which all lead to the first node of the community). The resulting chromosome gives the chromosome of the child.

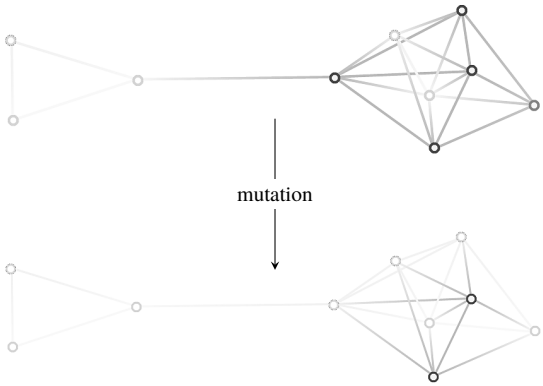


Fig. 6a: First mutation process
A random sample of nodes is assigned to random communities (e.g. the community of the red node in the above network is changed from red to green). Through this stochastic process we sample the solution space in an effective way. For example it enables the creation of new communities.

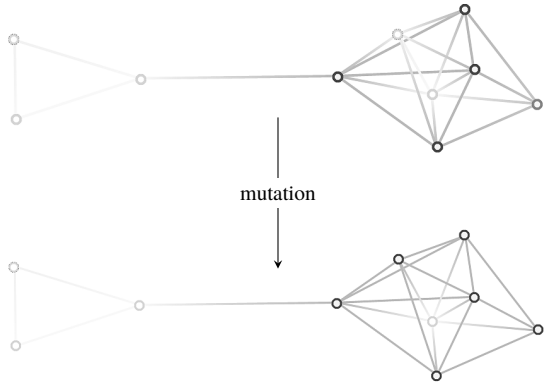


Fig. 6b: Second mutation process
A random sample of nodes inherits the most dominant community in its respective neighborhood (e.g. the red node in the above network inherits the purple community since three out of four neighbors are in the purple community). This improves the speed of convergence since it is likely that neighbors belong to the same community.

Finally the algorithm removes communities that consist of only one single node, because a proper community must contain at least two nodes. The whole process is repeated until a certain number of generations have been created. The algorithm then stops and returns the best found solution.

3 Experimental results

Since the algorithm finds solutions stochastically, its outcome might differ on every execution. To test the effectiveness of our algorithm two networks consisting of two respectively five communities were used (see fig. 7 and 8). For each trial the algorithm was run 100 times with the same parameters and the number of times where the algorithm found the correct community partition was counted.

We used a population size of 20 and, for each child, a 75% probability for the first mutation process and a 50% probability for the second mutation process. The algorithm was set to terminate after 5 generations for the first network and after 50 generations for the second network. For the *Mating*-method the 15% fittest were selected and all nodes from the selected communities were transferred (see fig. 5).

Results of the testing are shown in table 1. The two communities of the first network were correctly detected in all 100 runs within two generations on average. The communities of the second network were detected correctly in 95% of all cases for the *Elitism*-method and 99% for the *Mating*-method within eight respectively seven generations on average. In most of the failed cases the algorithm merged two communities into one single community.

The results show that our algorithm detects the community structure efficiently, as the average number of generations used to find the community structure was low. It also is effective, because the failure rate was very low as well (0% for the first network, 5% respectively 1% for the second network).

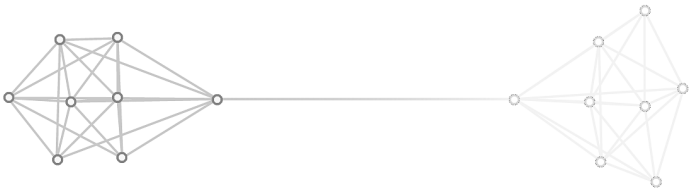


Fig. 7: First tested network consisting of 16 nodes, 49 edges and two communities. Number of possible community partitions $\approx 1 \cdot 10^{10}$

Also, the algorithm was tested using a large network with unknown community structure and its result were compared with different methods of *Mathematica*'s built-in function `FindGraphCommunities`. For our algorithm we used the *Mating*-method, the same parameters as for the other two networks and set it to terminate after 100 generations. In order to estimate

method	fails	avg. gen.	max. gen.
Elitism	0%	2.13 ± 0.34	3
Mating	0%	2.19 ± 0.39	3

(a) Results for first network

method	fails	avg. gen.	max. gen.
Elitism	5%	8.56 ± 5.43	30
Mating	1%	7.08 ± 2.53	27

(b) Results for second network

Tab. 1: Results of the testing
The columns show the used selection method, the number of runs where the algorithm did not find the correct community partition, and the average and maximum number of generations needed to find the best community partition.

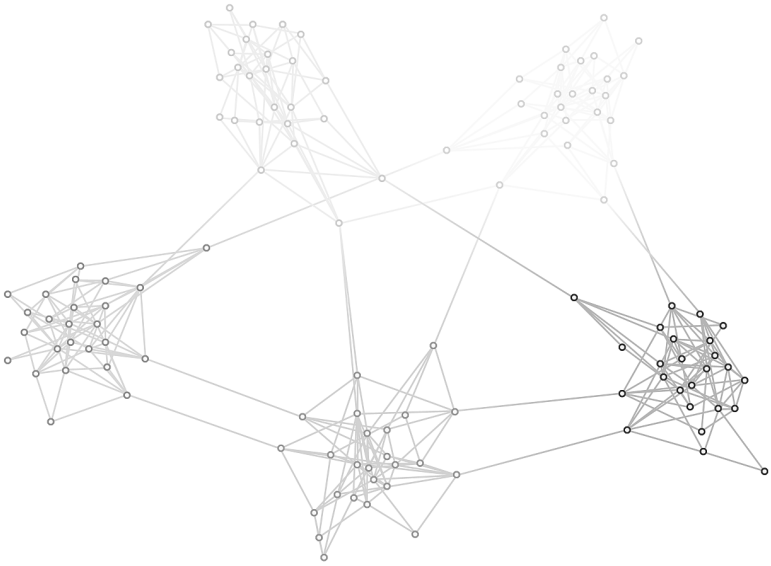


Fig. 8: Second tested network consisting of 125 nodes, 390 edges and five communities. Number of possible community partitions $\approx 2.5 \cdot 10^{153}$.

how far *Mathematica*'s results and the result of our algorithm were apart we calculated the partition distance, i. e. the number of nodes that were classified into different communities (this was transformed into an assignment problem and solved using *Mathematica*'s built-in function `FindIndependentEdgeSet`). Our algorithm detected 38 communities (see fig. 9). The community structure differs only slightly from the ones obtained by *Mathematica*'s *Centrality* and *Hierarchical* methods (see table 2).

The results of our algorithm yielded a higher modularity value than all different methods of *Mathematica*'s built-in function `FindGraphCommunities`. We are therefore tempted to claim that our algorithm is more effective. However, the outcome and hence effectiveness of our algorithm does not only strongly depend on the used parameters of our algorithm, but also the investigated network itself. Whether our algorithm outperforms *Mathematica*'s methods can hence not answered with certainty. In terms of speed our implementation of the algorithm can not compete with *Mathematica*'s function. Our implementation is almost 10^3 times slower (time-wise) and therefore not suited for networks consisting of millions of nodes.

Fig. 9: Large network consisting of 1293 nodes and 4145 edges showing the community structure detected by our algorithm. About $6.7 \cdot 10^{2609}$ possible community partitions.

method	communities	modularity	partition distance
Centrality	38	0.907235	3
Hierarchical	38	0.904835	8
Modularity	35	0.900042	83
Spectral	43	0.817467	143
Our algorithm	38	0.907236	-

Tab. 2: Comparison of the results of *Mathematica*'s built-in function `FindGraphCommunities` with the result of our algoritm. The columns show the method used by *Mathematica*'s built-in function, the number of detected communities, the modularity of the partition and the partition distance.

4 Analysis of the algorithm

Due to its stochastic way of finding solutions the algorithm does not always find the best solution, which is supported by the high degeneracy of the modularity function with a lot of local maxima close to the global maximum. However our simulations showed that it usually finds a solution that is at least close to the best solution.

Yet there might be a principal problem of the algorithm. Imagine n identical complete graphs K_m with m nodes each, which are arranged on a ring lattice and connected via a single link. Intuitively we would say that each complete graph forms a community since all nodes within a complete graph are densely (in fact maximally) connected and share only two links to nodes from other communities. However it was proved that for a sufficiently large number of graphs n modularity maximization will lead to a partition that merges pairs of communities into one single community [FB07]. Therefore modularity maximization can lead to a partition which seems counter-intuitive as it "neglects" small community structures, which is however the consequence of the implicit definition of a community using modularity (see fig. 10). In general, communities smaller than a certain threshold, which depends on the network size, will always be merged, which is known as *resolution limit*. A possible way to solve this problem is to check every community whether it is a merging of smaller communities.

5 Conclusion

In this article we proposed a genetic algorithm aimed at detecting the community structure of a network by maximization of a measure called *modularity*. The algorithm consists of different genetic operators, namely a selection process, a cross-over process and a mutation process. It provides two different methods for the selection process. Tests using several sample networks of known community structure showed that our algorithm detects communities both efficiently and effectively. For the investigated network with unknown

Fig. 10: Resolution Limit
Example network consisting of 10 complete graphs with three nodes each. For the left, intuitively expected community partition the modularity is 0.65, whereas in the right partition, where each time two communities are merged, the modularity is 0.675, which is higher than for the expected partition. Modularity maximization will therefore lead to a community partition that might be counter-intuitive.

community structure the algorithm even seems to outperform *Mathematica*'s built-in methods for finding communities. Hence we can conclude that our algorithm shows a very good performance in detecting the communities of complex networks and is suited for cases where a community partition close to the best community partition is required. However, our algorithm can fail to resolve sufficiently small communities as intuitively expected due to a problem known as *resolution limit*. From the investigation of the fitness function *modularity* the question arises if one could improve the definition of modularity such that it also resolves sufficiently small communities as intuitively expected.

References

[Ba16] Barabási, A.-L.: Network Science. Cambridge University Press, 2016.

[FB07] Fortunato, S.; Barthélemy, M.: Resolution limit in community detection. PNAS volume 104 no. 1/, pp. 36–41, 2007.

[FC12] Fortunato, S.; Castellano, C.: Community structure in graphs. Springer New York Computational Complexity: Theory, Techniques, and Applications/, pp. 490–512, 2012.

[FH16] Fortunato, S.; Hric, D.: Community detection in networks: A user guide. Physics Reports 659/, Community detection in networks: A user guide, pp. 1–44, 2016.

[Fo10] Fortunato, S.: Community detection in graphs. Physics Reports 486/3, pp. 75–174, 2010.

[HTF09] Hastie, T.; Tibshirani, R.; Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction, Second Edition. Springer New York, 2009.

[KN11] Karrer, B.; Newman, M. E. J.: Stochastic blockmodels and community structure in networks. Phys. Rev. E 83/, p. 016107, Jan. 2011.

[Le18] Lehnerer, S.: SimonNick/genetic-algorithm-community-detection: Initial release, 2018, URL: https://zenodo.org/badge/latestdoi/139072714.

[Lu07] Luxburg, U.: A Tutorial on Spectral Clustering. Statistics and Computing 17/4, pp. 395–416, Dec. 2007.

[Ne06] Newman, M. E. J.: Modularity and community structure in networks. Proceedings of the National Academy of Sciences 103/23, pp. 8577–8582, 2006.

[Po97] Pothen, A.: Graph Partitioning Algorithms with Applications to Scientific Computing. In (Keyes, D. E.; Sameh, A.; Venkatakrishnan, V., eds.): Parallel Numerical Algorithms. Springer Netherlands, Dordrecht, pp. 323–368, 1997.

[RB08] Rosvall, M.; Bergstrom, C. T.: Maps of random walks on complex networks reveal community structure. Proceedings of the National Academy of Sciences 105/4, pp. 1118–1123, 2008.

[Sc07] Schaeffer, S. E.: Survey: Graph Clustering. Comput. Sci. Rev. 1/1, pp. 27–64, Aug. 2007.

Multimedia und Datenverarbeitung

The Omniscope - Multimedia Streaming and Computer Vision for Applications in the Virtuality Continuum

Gerald Melles[1]

Abstract: Researching applications within the Virtuality Continuum (VC) is a process involving combinations of many different technologies. Media streaming and computer vision in particular are important aspects of many VC applications. This paper introduces the Omniscope library as a way to integrate both in an efficient, user-friendly and extensible manner. It achieves this by combining GStreamer and OpenCV in a C/C++ library as well as a plugin for the Unity IDE.

Keywords: virtuality continuum; VR; streaming; computer vision; OpenCV; GStreamer; Unity

1 Introduction

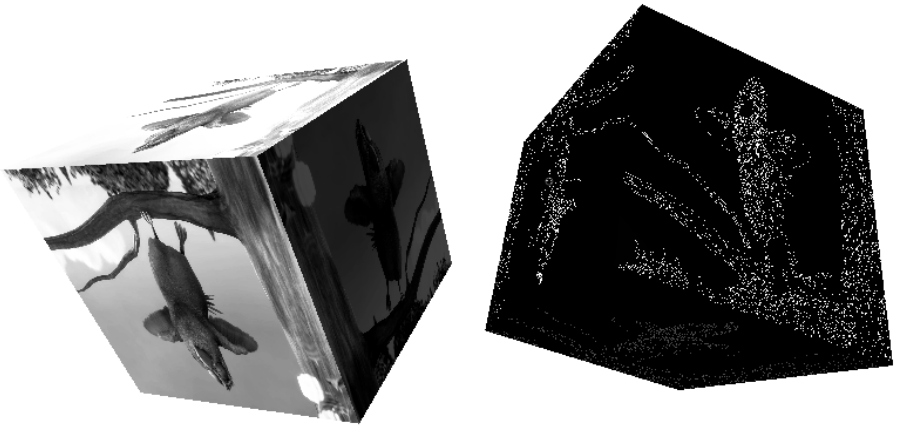


Fig. 1: Omniscope Plugin rendering and transforming a Full-HD video using edge detection

Multimedia streaming is a common requirement in systems within the virtuality continuum (VC) which encompasses Augmented, Virtual, Mixed and Blended Reality (cf. [MK94]). Among other aspects, media streams are often used to transport the required data for visual and auditory stimuli to and from peripheral devices.

[1] Hamburg University Of Applied Sciences, Fakultät TI-I, Berliner Tor 5, 20099 Hamburg, Deutschland
contact@geraldmelles.com

VC applications also make extensive use of computer vision algorithms, such as in camera-based tracking and motion capture systems. In [Ng17] for instance, Nguyen et al. propose a new and less obtrusive design for markers similar to QR-Codes and suggest their use in VR applications.
VC research projects often need combinations of both of these aspects, for example to take the user's gaze into account to render in a foveated manner, such as in [Lu17]. There is no lack of projects and proposals in this intersection, but their tooling tends to be either purpose-built for a specific use case or proprietary.
Generic multimedia streaming and computer vision libraries exist, in both proprietary and open source variants. The main challenge lies in their integration into the game or simulation engine at the core of the VC application. This is exacerbated by requirements concerning speed, efficiency and extensibility, the maintenance of platform independence and the support for a range of data formats and devices. This may go some way toward explaining why there is no generic and open solution. Computer vision algorithms in particular often remain inaccessibly embedded within proprietary software and hardware products. Rapidly prototyping a VC application combining the two as well as maintaining the ability to change their internal workings is not supported by software solutions that are available today.
The Creative Space for Technical Innovations (CSTI) at the Hamburg University of Applied Sciences (HAW) primarily uses the Unity IDE as the simulation engine for developing their applications. This proprietary IDE is one of only a few viable options for the development of state-of-the-art VC systems, especially when developing a new simulation engine is not an option (i.e. due to the time and effort required). It can be extended through scripting and plugins in a variety of programming languages. At the time of writing, Unity is one of the most commonly used IDEs for VR and AR simulations, with its main competitors being other proprietary products like the Unreal Engine and Cryengine. In part, this dominance of proprietary engines may be related to the need to support the peripherals and platforms unique to VC systems, such as head-mounted displays, wands and motion tracking systems, which are also proprietary themselves.
Both the IDE's built-in media streaming support and its open source plugins lack a way to efficiently interface them with computer vision libraries, in addition to providing only very limited support for formats and encodings. There are also no comprehensive open source integrations of computer vision libraries themselves.
This paper presents the Omniscope library and plugin for Unity as a means of making the research and development of media streaming and computer vision based applications in the virtuality continuum faster, easier and more efficient. It was developed as a link between the two fields: A C/C++ library combining the GStreamer and OpenCV libraries. It is intended to make media streaming and computer vision features as accessible and customizable as possible for researchers into VC systems at the CSTI and elsewhere and can be integrated into Unity using the IDEs native plugin architecture. Plans exist to adapt it for other VC development environments, such as the Unreal Engine.
Fig. 1 shows the Plugin rendering an MP4 video[Bi08] onto cubes in a Unity scene. On the right cube, an edge detection algorithm was applied before rendering (an implementation of the algorithm proposed by John Canny in [Ca86]).

2 Architecture

Audio and video streams flow back and forth between servers, the VC simulation system and its various peripheral devices. A large number of standards, formats and encodings are in use, both free and proprietary. A few streaming frameworks seek to support as many of these as possible, both by endeavoring to be agnostic to the nature of the data they handle and by utilizing plugin architectures for components like encoders/decoders to ease their integration. One such framework is GStreamer, which is largely platform-independent and included in most Linux distributions.
OpenCV is a well-established computer vision library. Its open source nature has made it a popular basis for research into computer vision, particularly of feature detection algorithms. While it does already offer an integration with GStreamer, the Omniscope library instead offers its own in order to allow for better extensibility (i.e. other streaming framework integrations) and more complete playback support.
The Omniscope plugin links with both GStreamer and OpenCV dynamically. This is largely to accommodate the possibility of the user requiring all of the libraries' modules to have a specific (e.g. non-proprietary) licensing model and to enable its use in applications without licensing issues.[2]
Similarly, the plugin's functionality is exported to Unity using C-style bindings in a dynamically loaded library.[3]
The plugin's pipeline architecture with elements, sources and sinks is similar to that of GStreamer. Like GStreamer, the Omniscope library also largely abstracts from its elements' implementations. This way, elements are also more easily added to or removed from a project, regardless of their dependencies.
The Omniscope project currently consists of six C/C++ subprojects: Five internal modules and a standalone application. These subprojects are intended to be built using the GNU GCC Compiler (cf. [GN18]) for Linux-based systems or cross-compiled for Microsoft Windows using MinGW-w64 (cf. [Mi18]). Additionally, the project contains C# scripts for easier access to the library from within the IDE.

- Omniscope Common Module
- Omniscope GStreamer Module
- Omniscope OpenCV Module
- Omniscope Core Module
- Omniscope Unity Module
- Omniscope Standalone application

[2] Note that all GStreamer plugins used by the Omniscope plugin are licensed under the LGPL and OpenCV under the 3-clause BSD license.
[3] The Omniscope Unity module itself contains some code by Unity Inc., under the MIT license.

Omniscope Common contains header files with the pipeline elements' base interfaces and a custom thread pool implementation.
Omniscope GStreamer contains element specializations for GStreamer pipelines and elements. It provides support for GStreamer's gst-launch command line syntax, which is the recommended way for the user to extend the Omniscope plugin's streaming functionality without having to resort to working with its source code. It is required that the user supply at least one appsink or appsrc element in the pipeline definition as these are used for the exchange of samples between GStreamer and other pipeline elements.
Omniscope OpenCV consists of elements offering stream capture and processing implementations using OpenCV. It can be extended with custom sample analysis and processing modules. In its simplest form, a new sample processor can be implemented by providing a new override to a single function taking and returning a generic media sample instance (which may contain several frames and additional information). By default, sample processing functions are automatically executed asynchronously using a thread pool.
The Omniscope Core provides the pipeline itself: an API for instantiating and connecting sources, sample processors and sinks, manipulating their state (e.g. playing, pausing and seeking) and accessing the resulting media samples and analysis results.
The Omniscope Unity module adapts Unity's low-level native rendering plugin support (in C++) to permit rendering captured and optionally processed frames directly to existing target textures. This low-level access to Unity's rendering APIs makes the Omniscope much faster than passing frames up to Unity's C# scripts for rendering would be. It is also capable of interfacing with different graphics architectures and maintains the same platform independence as the rest of Omniscope's modules. In addition, any analysis results of sample processors can also be accessed through it. It is nevertheless recommended to use the provided C# scripts to facilitate communication between the Omniscope library and Unity. These use the Unity IDE's component system to provide a graphical UI (cf. 'Usage'). The project also contains a simple standalone application built upon the other modules (excluding the Omniscope Unity module). This primarily serves as a means to ease development and testing of new features but could also be used for the development of standalone streaming and computer vision applications.

3 Usage

Most of Omniscope's complexity can be hidden behind a graphical UI. The user may interact with the plugin in four ways (ascending by flexibility and descending by ease of use):

- Using Omniscope's UI (in Unity)
- Using Omniscope's C# API (within Unity's scripting system)
- Using Omniscope's 'C' style linkages
- Extending Omniscope's modular architecture in C/C++

The first option consists of an extension to Unity's component inspector. It offers input fields and buttons (no programming is required) and is intended to be used primarily for simple audio and video playback and the use of pre-built frame processors. The use of GStreamer's gst-launch syntax adds additional flexibility.
The second option is intended for developers who either want to have programmatic control over the plugin's behavior at runtime (e.g. for pausing playback or seeking) or to use the built-in computer vision algorithms for frame-by-frame analyses. For example, it could be used to apply a feature detection algorithm as a sample processor and receive the relative positions of detected features (such as eyes and faces) once per render cycle.
If the C# API does not suffice or if the plugin is to be used outside the Unity IDE, the user may also use the plugin's 'C' style linkages.
Lastly, as mentioned above, the Omniscope's architecture has been designed with extensibility in mind. Its central features are abstracted through interfaces and abstract base classes, easing the development of new elements, such as frame processing or stream capture implementations.
The Omniscope's focus lies on video and vision, but it also offers limited support for other media such as audio streams and subtitles to accompany video playback.

4 Conclusion

The need for well-integrated, efficient, open and extensible streaming and computer vision support for the research and development of VC applications has been established. The Omniscope project has been introduced as a solution, combining the GStreamer and OpenCV libraries and optionally integrating them with the Unity IDE. Its uses in VC applications include playing back multimedia streams from a variety of sources (both video and audio), transforming the visual frames - for example to show detected edges in a stylized way - and extracting other data through computer vision algorithms, such as the location of detected faces or tracking markers. It is being used in the CSTI and steadily improved and extended.

5 Further Research

While the Omniscope is already in use by researchers in the CSTI, formal and exhaustive analyses of its speed and efficiency have yet to be performed.
The Omniscope could also serve as a means of integrating other media based features with VC applications, such as three-dimensional audio processing based on the acoustics of a virtual space. Related work is being done at the HAW's wave field synthesis lab, for example exploring the use of acoustics in redirected walking (cf. [NF16]).
The capture of multimedia streams and their processing by computer vision algorithms are also only some of the aspects of networked VC systems. These systems' communication with various peripherals, local software and hardware landscapes and remote services provides many other challenges worthy of research. Of particular interest to researchers in the CSTI

is the integration of embedded systems and smart environments in VC systems. To this end, a number of projects have sprung up, such as the CSTI middleware (cf. [Ei17]).

References

[Bi08] Big Buck Bunny, original by the Blender Foundation (peach.blender.org), official mirror by Janus B. Kristensen (http://bbb3d.renderfarming.net), last accessed 01.07.2018).

[Ca86] Canny, J.: A Computational Approach to Edge Detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-8(6):679–698, Nov 1986.

[Ei17] Eichler, Tobias; Draheim, Susanne; Grecos, Christos; Wang, Qi; von Luck, Kai: Scalable Context-Aware Development Infrastructure for Interactive Systems in Smart Environments. In: Fifth International Workshop on Pervasive and Context-Aware Middleware 2017 (PerCAM'17). Rome, Italy, October 2017.

[GN18] GNU Compiler Collection (official site): https://gcc.gnu.org/.

[Lu17] Lungaro, Pietro; Tollmar, Konrad; Mittal, Ashutosh; Valero, Alfredo Fanghella: Gaze- and Qoe-aware Video Streaming Solutions for Mobile VR. In: Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology. VRST '17, ACM, New York, NY, USA, pp. 85:1–85:2, 2017.

[Mi18] MinGW-w64 - GCC for Windows 64 & 32 bits (official site): https://mingw-w64.org.

[MK94] Milgram, Paul; Kishino, Fumio: A Taxonomy of Mixed Reality Visual Displays. In: IEICE Trans. Information Systems. volume vol. E77-D, no. 12, pp. 1321–1329, 12 1994.

[NF16] Nogalski, M.; Fohl, W.: Acoustic redirected walking with auditory cues by means of wave field synthesis. In: 2016 IEEE Virtual Reality (VR). pp. 245–246, March 2016.

[Ng17] Nguyen, Minh; Tran, Huy; Le, Huy; Yan, Wei Qi: A Tile Based Colour Picture with Hidden QR Code for Augmented Reality and Beyond. In: Proceedings of the 23rd ACM Symposium on Virtual Reality Software and Technology. VRST '17, ACM, New York, NY, USA, pp. 8:1–8:4, 2017.

Fundamentals of Real-Time Data Processing Architectures Lambda and Kappa

Martin Feick, Niko Kleer, Marek Kohn[1]

Abstract: The amount of data and the importance of simple, scalable and fault tolerant architectures for processing the data keeps increasing. Big Data being a highly influential topic in numerous businesses has evolved a comprehensive interest in this data. The Lambda as well as the Kappa Architecture represent state-of-the-art real-time data processing architectures for coping with massive data streams. This paper investigates and compares both architectures with respect to their capabilities and implementation. Moreover, a case study is conducted in order to gain more detailed insights concerning their strengths and weaknesses.

Keywords: Software architecture, Big Data, real-time data processing, Lambda and Kappa architecture

1 Introduction

The internet is a global network that is becoming accessible to an increasing number of people. Therefore, the amount of data available via the internet has been growing significantly. Using social networks for building communities, distributing information or posting images represent common activities in many people's daily life. Moreover, all kinds of businesses use technologies for collecting data about their companies. This allows them to gain more detailed insights regarding their finances, employees or even competitiveness. As a result, the interest in this data has been growing as well. The term *Big Data* is used for referring to this data and its dimensions.

As Big Data has progressively been gaining importance, the need for technologies that are capable of handling massive amounts of data has emerged. In this paper, one technology of interest is the so-called Lambda architecture that was introduced by Nathan Marz in 2011 [Ma11]. Its introduction was motivated by the purpose of beating the popular CAP theorem [Br00]. The CAP theorem states that a shared distributed data system is incapable of guaranteeing Consistency, Availability and Partition tolerance at the same time. Instead, only two constraints can at most be enforced. Marz emphasizes that the architecture does not rebut the CAP theorem but simplifies its complexity to allow for more human-fault tolerance when developing shared data systems.

The Lambda architecture has enjoyed broad attention which has led to numerous people

[1] Hochschule für Technik und Wirtschaft des Saarlandes, Fakultät für Ingenieurwissenschaften, Goebenstraße 40, 66117 Saarbrücken, Deutschland; e-mail: {mfeick,nikleer,mkohn}@htwsaar.de

sharing their thoughts about the technology. A considerably influential article by Jay Kreps (2014) acknowledges Marz's contribution for raising awareness about commonly known challenges of building shared data systems. At the same time, he discusses some disadvantages of the Lambda architecture [Kr14]. Consequentially, he proposes an alternative real-time data processing architecture, termed Kappa architecture, as an alternative. In contrast to the Lambda architecture, Kreps's approach is supposed to simplify any development related matters.
In this paper, we investigate the Lambda as well as the Kappa architecture, take a more detailed look at their functionalities and compare their capabilities. Therefore, the paper is divided into the following sections. Subsequently, we elaborate on the related work regarding Big Data and real-time data processing. After that, we take a more detailed look at both architectures including their workflow and implementation. Moving to section 5, we conduct a case study in which we compare both architectures[2]. This way, we are be able to analyze each architecture's strengths and weaknesses more effectively. A subsequent discussion proceeds by emphasizing significant details regarding our results. Finally, we conclude this paper's results and consider potential future work in the last section.

2 Related Work

In the related work section, we first introduce the term Big Data, and we briefly discuss the issues related to it. Afterwards, we look at *data processing solutions* as well as the term *data analytics* in order to support real-time Big Data streams.

2.1 Big Data

Over the last decade, the term Big Data became more and more relevant. Big Data has its place in almost every business area such as information technology, healthcare, education etc. However, the term Big Data does not only cover the pure size of data [Ma15, Ma11]. Instead, Big Data is composed of three standard dimensions known as the three V's, which mean *Volume*, *Variety*, *Velocity* [Ga15]. Additionally, certain companies contributed other dimensions to Big Data. For example, IBM added *Veracity* as a forth V, SAS introduced *Variability and Complexity*, and finally Oracle brought up *Value* [Ga15].
Often, the scale of data needed to support the various application scenarios is too big for a traditional database approach. Handling such an amount of data cannot be done by simply increasing the resources, because it does not consider the higher complexity and coherence of the data [Ma15, Ga15]. The next section introduces real-time data processing solutions considering all previously introduced dimensions of Big Data.

[2] The project is available on GitHub: `https://github.com/makohn/lambda-architecture-poc`

2.2 Real-Time Data Processing Solutions

Hasani et al. [Ha14b] outline that particularly the *Velocity* aspect of Big Data is difficult to handle effectively. Technologies must be able to handle real-time stream processing at a rate of millions per second. Furthermore, data streams can be collected from various sources using parallel processing. However, the goal of Big Data is to gain knowledge about the data and this is only attainable with the help of data integration and data analytics methods [Li14]. Data analytics is essentially the process of examining a data set and conclusively getting the insight/value of the data [Li14]. However, for traditional tools, it is challenging as soon as the data size extensively grows [Ha14a]. In addition, besides the most recent data, for some requests all the data is needed as it leads to more accurate outcomes [Ki15, Ha14a]. As a result, accessing information within a time limit is often not possible due to the size of the data [Li14].
A common strategy to face this challenge is to use hybrid techniques [Ki15, Ma15, Ha14a]. Abouzied et al. [Ab10] discussed HadoopDB which combines MapReduce and DBMS technologies. It is used to analyze massive data sets on very large clusters of machines [Ho12]. They present different real world applications e.g. a semantic web data application for protein sequence analysis [Ab10]. Their results show that HadoopDB is an effective platform for retaining a large data set and performing computation on it. However, HadoopDB has its limitations when using real-time data streams [Li14].
We previously introduced two general requirements of Big Data systems. First, receiving a massive real-time data stream from different sources and second, performing an analysis of this data in order to output results almost immediately [Ma15, Li14, Ha14b, Ki15]. From this point, we move on to two concrete software architectures/patterns called Lambda and Kappa that are nowadays commonly used for Big Data systems.

3 Lambda Architecture

The Lambda architecture has been given its name by Nathan Marz [Ma11], and describes a generic, scalable and fault-tolerant real-time data processing architecture. It provides a general-purpose approach to apply an arbitrary function on an arbitrary data set [Ma11]. Marz defines the most general-purpose function, as a function that takes all the existing data as input (*query = function(all data)*), and returns its results with low latency. However, calculating results to ad-hoc queries using the entire data set is computationally expensive. Therefore, the Lambda architecture uses pre-computed results (views) being able to respond with low latency [Ma15].
Figure 1 shows an overview of the Lambda architecture comprising three layers. The batch layer has essentially two functions: (1) It stores an immutable master data set and (2) is responsible for pre-computing the batch views based on this data set. The speed layer is responsible for indexing real-time views, compensating the high latency of the batch layer. In particular, due to the massive data sets in the batch layer, it takes time for the latest batch layer views to be calculated, causing a lack of availability. The speed layer is used to close

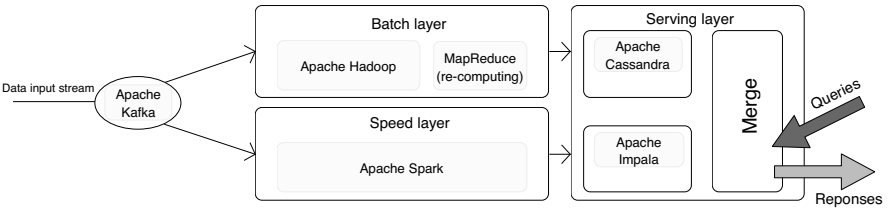


Fig. 1: The Lambda architecture and its workflow through Batch, Speed and Serving layer

this gap by providing an efficient way for querying most recent data [Ma15]. As soon as the batch layer has re-computed its views, the speed layer discards the redundant data, and hence they are provided by the batch layer views. Moreover, there are queries where most recent data and data from the batch layer are required. Therefore, the serving layer merges results from batch and speed layer views. Further, the serving layer takes care of indexing and providing the merged views, enabling easy access for the user.

3.1 Workflow & Technologies

As illustrated in Figure 1, all incoming data is dispatched to batch and speed layer for further processing. Since we talk about a real-time stream processing with a massive amount of data, a common technology to realize this is Apache Kafka [Du16].
Moving on to the batch layer, a standard technology to store the master data set and to perform the recomputations of the batch views is Apache Hadoop [Ha14a]. Hadoop is an open-source framework that ”allows the distributed processing of large data sets across clusters of computers using simple programming models” [Ho12]. Following the general application domains and purpose of the Lambda architecture, the master data set grows extensively over time [Ma11]. MapReduce allows the system to compute the batch views even on a large data set [Ho12]. It is composed of three steps (Map, Shuffle, Reduce) using clusters for distributed parallel computations [De08]. MapReduce aims to parallelize the mapping, shuffling and reducing steps in order to significantly improve the time complexity of computations on large data sets [De08]. Notice, that by the time the batch layer views are generated, they are already outdated as a result of the sustained real-time stream processing [Ma11]. This leads us to the speed layer that compensates the high latency of the batch layer and provides the recent data only. For instance, Apache Spark is used to implement this layer in order to reach the required performance [Ha14b]. Spark is an engine particularly developed for large-scale Big Data processing. Spark maintains Apache MapReduce’s linear scalability, and fault tolerance while improving its performance considerably.
Finally, the serving layer stores batch and speed layer views, and subsequently responds to ad-hoc queries by returning the pre-computed views. We distinguish between requests

addressed to views from batch and speed layer, and others that require the use of views from both layers simultaneously. To respond to such requests the serving layer must merge different views (see Figure 1). Generally, the amount of data in the serving layer is relatively small as it only hosts the computed views from batch and speed layer. A common technology for storing batch views is Apache Cassandra views [Ha14a]. Cassandra is a NoSQL database featuring a distributed deployment providing a high level of reliability [Ne13].

3.2 Trade-offs

Using the Lambda architecture has various advantages such as fault-tolerance against hardware failures and human mistakes. It also addresses the problem of computing arbitrary functions on arbitrary data in real-time [Ma15]. However, the software architecture pattern is highly complex and redundant. In oder to apply the Lambda architecture for a specific use case, it has to be tailored correspondingly. Moreover, the different technologies that are needed to run batch, speed and serving layer make it challenging to implement (see Figure 1). Furthermore, keeping both, batch and speed layer synchronized, increases the computational time and effort. In addition, maintaining and supporting both layers is difficult because they are distinct and fully distributed [MJ17].
In summary, the Lambda architecture achieves its goals but comes with high complexity and redundancy. The question arises whether the majority of the use cases require a batch and a speed layer or not. Before we move on to this specific question in our case study, we introduce the Kappa architecture.

4 Kappa Architecture

The Lambda architecture enjoyed comprehensive attention after it was introduced by Marz [Ma11]. Only a few years later, Jay Kreps, a principal staff engineer at LinkedIn, shared his thoughts about the Lambda architecture pointing to its naturally existing disadvantages [Kr14]. Kreps presented another approach for real-time data processing that, in contrast to the Lambda architecture, favors simplicity with respect to development related matters – the Kappa architecture. In this section, we take a closer look at the architecture's components, their functionality and point to several similarities as well as differences compared to the Lambda architecture.

Fig. 2: The Kappa architecture and its workflow through Real-Time and Serving layer

4.1 Overview

The Kappa architecture represents another way of designing a stream processing system. Similarly to section 3, we start off by introducing the architecture's components and elaborate on their functionality. Figure 2 provides the basic outline of the Kappa architecture. Once again, the architecture requires a data stream. First, the incoming data is fed into a stream processing system, sometimes referred to as the real-time layer [Zh17]. This layer is responsible for running the stream processing jobs and providing real-time data processing. Afterwards, the data is directed into the serving layer that queries any required results. Notice that this architecture's components do not particularly differ from the Lambda architecture. Moreover, there is no need to elaborate on any further technologies that are used for implementing this architecture. That is because the Kappa architecture can be implemented by using the same open source technologies previously presented for realizing the Lambda architecture, as illustrated in Figure 2. Next, we take a closer look at the architectures differences.

4.2 Distinguishing Lambda and Kappa

Even though there are numerous similarities, considerable differences arise from the fact that the Kappa architecture passes on a batch processing system. This way, the architecture only requires one code base instead of implementing two heterogeneous systems [OA16]. As a result, development related processes like implementation, debugging and code maintenance are simplified. On the other hand, passing on a batch layer also results in the architecture to be incapable of managing computation intensive applications. This is the case with respect to large scale machine learning scenarios where a model needs to be trained [Zh17]. Furthermore, the performance of batch processing tasks in general suffers from the unavailability of a batch layer [Li17]. However, the Kappa architecture's disadvantages are not particularly problematic as Kreps suggested this approach as an alternative to the Lambda architecture valuing simplicity over efficiency [Kr14]. This means that a direct comparison of both architectures is difficult since the performance of the architecture largely depends on the use case. Therefore, the most appropriate architecture always has to be chosen based on the given application scenario.

5 Case study

In order to provide a comprehensive overview on how both, the Lambda architecture and the Kappa architecture, are implemented, we conduct a case study in the following section. Comparing both architectures, we investigate a stereotypical use case, pointing out advantages and challenges. Based on the technologies presented in subsection 3.1, we develop a *proof-of-concept* implementation. In doing so, we explain the individual technologies in more detail and explain why they are used in the particular case. Ultimately, we try to give an extensive overview of the Lambda architecture, while constantly keeping in mind the approach of the Kappa architecture, investigating structural differences.
As data source, we use Twitter's streaming API, which provides us with comprehensive data about tweets and users. These contain both unstructured data (a tweet's text) as well as structured metadata about the tweet (the tweet's ID, timestamps or included hashtags). Using this data as an input, we are aiming to analyze the hashtags according to their popularity. For the development of our software we use the multi-paradigm programming language *Scala* since it allows smooth integration of the above mentioned tools. Furthermore, thanks to its functional approach, Scala comprises a number of integrated functions allowing it to seamlessly implement technologies such as MapReduce [Up17].

5.1 Providing the data

To access the data of the Twitter streaming API, we use the Twitter4J library. This requires a corresponding registration of the app on Twitter and allows us to access a filtered stream using OAuth authorization. We use a location-based filter that uses minimum and maximum values of longitude and latitude as its range allowing us to access a broad spectrum of all tweets. Thus, we receive a large amount of tweets in very short periods of time, impeding the immediate processing of tweets as they come in. As mentioned earlier, it is reasonable to delegate the buffering of messages to a message broker, as for instance Kafka. This is mainly due to its asynchronous and message-based communication which also implies a complete decoupling of senders and receivers [Du16]. In our example, we utilize the Producer API to write tweets into a queue when they arrive and the Consumer API to read from the queue to populate batch and speed layers. Note that Kafka is usually implemented as a cluster and therefore multiple bootstrap servers can be specified to host this cluster. A cluster node, also referred to as a broker, is responsible to store messages within a specified topic. This topic is unique and can be subscribed by various consumers. In order to allow for parallelism, especially when using Kafka as a distributed cluster, topics are further divided into partitions. In order to take the sequence of incoming messages into account, each message is annotated with a timestamp. This way, messages from different partitions can later be merged together easily [Du16]. For each tweet we consider the set of hashtags. In order to enable a clear identification later on, each message sent to Kafka contains not only the hashtag's text but also the tweet's ID as well as its user's screen name and its timestamp. Since we prefer a serialized yet object-oriented format for message exchange,

we convert every Hashtag object into the JSON format. This allows for high compatibility with Cassandra.

5.2 Implementing the batch layer

Now that we have provided a stream of messages, we can start processing them. First, we need to implement a consumer in order to read messages from the Kafka queue. This consumer is primarily responsible for filling the master data set. The idea here is that data is not accidentally changed or deleted, which results in a high degree of consistency [Ma15]. Based on this data, specific views are later calculated to display concrete information (such as the number of hashtag occurrences). In order to achieve a realistic processing speed, we implement the consumer in such a way that it reads multiple messages at a time from the Kafka message queue, writing them to the Cassandra database. This process is scheduled to be executed at a regular interval. The scheduling is done by implementing the consumer as an *Actor*. Actors are a basic concurrency construct in Scala, somewhat comparable to tasks in other programming languages, with the difference that actors can communicate with each other [Up17].
As previously mentioned, we want to use Cassandra as the database of our choice. This enables SQL-like queries that can be created using CQL (*Cassandra Query Language*) [Gu16]. CQL supports all common CRUD operations. As it is possible to assign tables to certain namespaces, called *keyspaces*, we define three different key spaces, for the master data set, the batch view and the realtime view. This allows us to create tables of the same name to enable uniform access to batch and speed views.
As the master data set now gets populated with new hashtags at a regular interval, we can start calculating batch views. Again utilizing a scheduler, we execute a batch job, which iterates through the whole data set, regularly counting occurrences of same hashtags. As the database grows over time, it is inadequate to sequentially count the occurrences as it is done in the speed layer. Instead, it might be reasonable to apply concurrent methods, ideally within a distributed system. One such method, MapReduce, has already been presented in subsection 3.1. After retrieving a list of hashtags from the master data set, we can distribute equally sized chunks of them to several map processes [Gu15]. Each map process then emits a key-value pair, mapping the hashtag as a key to an initial value of 1. Next, while implicitly shuffling same-titled hashtags to dedicated chunks, the reduce function sums up the values of same-titled hashtags. This way we now receive a new list of key-value pairs with a hashtag as the key and the number of that hashtag's occurrences in the data set as the value [Gu15].

5.3 Implementing the speed layer

While the batch layer works on the basis of the immutable master data set, the speed layer receives the stream of new data as an input. Therefore, the results of the speed layer represent only a sample of the total amount of data. Considering the Kappa architecture, the results of

the batch layer can be approximated by interpreting a batch as a limited stream. Here, one does not define a batch as a function on the entire data set, but rather as a function on an arbitrarily large recording of the stream.
In contrast to the batch layer, the retrieved data is not written to a database, but is forwarded directly to a calculation unit. This can be achieved by using Apache Spark, especially by leveraging a data structure called the *Resilient Distributed Dataset* (RDD) [KW17]. This is basically an immutable collection of data records that might reside on multiple nodes in a cluster. Each operation on a RDD requires the construction of a new RDD, memorizing the resulting hierarchy in the *RDD lineage graph*. This allows for fast computation, as data is kept in memory. Further, the concept of a `DStream` represents a continuous flow of RDDs, each representing a fixed windows of data received from the stream. A `ViewHandler` is given a `DStream`, allowing it to continuously executing fast calculations on small data chunks. Figure 3 illustrates the operating principle.
In the ViewHandler we now convert the RDD into a so-called `DataFrame`. This is a kind of wrapper that allows us to execute SQL-style queries on the RDD. The necessary methods and concepts are included in the module SparkSQL. This allows us to apply aggregate functions to the data. In particular, by grouping the hashtags, we can assign them with the number of their occurrences. Note that this is in general executed sequentially. Therefore –and in contrast to the batch layer– it is inapplicable for larger data volumes. As in the batch layer, the resulting hashtag-count pairs must be timestamped, allowing both views to be combined later on.

5.4 Implementing the serving layer

Now that we are able to perform calculations on both, batch and speed layer, we need to consider how to provide the results to the user. While being responsible for providing an easy-to-use interface for queries, the serving layer is also in charge of merging the results from the individual layers. Hence, if you want to have an exact assertion about the number of hashtags at a certain point in time, it is inevitable to compensate the batch layer's calculation latency by merging the results from the speed layer. Considering Table 1, one can see that there are overlaps of hashtags in batch view and real-time view. However, since the results of the speed layer were retrieved shortly after a football match, the hashtags correspond to the football match. For creating an interface, we use Akka to create a http server with a RESTful API, providing ordered JSON lists of hashtags as a result for queries. The

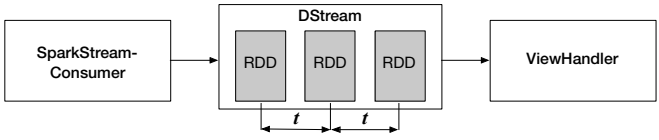


Fig. 3: Principle of the `DStream` in conjunction with a `ViewHandler`

Batch layer results (24 hours)		Speed layer results (1 hour)	
hashtag	**count**	**hashtag**	**count**
job	25970	FRAARG	2268
CareerArc	23256	job	1851
Hiring	19835	ARG	1643
YksBirincisiEmreP.	15614	FRA	1602
FRAARG	9813	CareerArc	1550

Tab. 1: Top 5 Hashtags in the batch view and in the realtime view after reading the twitter stream for 24 hours and 1 hour, respectively

merging is performed in Cassandra using tailored CQL queries. In particular, we consider the timestamp of our data when querying the data set. We select the batch view with the latest timestamp, storing the results into a list of hashtag objects. Further, we select all real-time views having a more recent timestamp than the batch view, also storing them into a list. We can then merge both lists in order to retrieve a new list comprising the updated hashtag objects.

6 Discussion & Limitations

While the Lambda architecture allows both high accuracy and fast processing of requests, one does this at the cost of maintaining two separate code bases and hence two complex, distributed systems. This results in some difficulties. On the one hand, both layers must be kept synchronous. If you change a particular view in one layer, the corresponding view must be adapted in the other layer as well. Further, merging in the service layer involves a certain complexity. The data must be structured in a way that efficient merging is possible. Thus, designing the database schemes to be compatible with each other is essential. Moreover, there must be a feature that allows the comparison of the data sets, such as timestamps. In addition, as the master data set grows, more hardware resources are needed in order to compensate the increase of latency while performing batch calculations.
The Kappa architecture, on the other hand, does not integrate a dedicated batch layer at the expense of accuracy. This is based on the assumption that numerous applications do not require the entire data volume, but a sufficiently large segment of the current streaming data. Nevertheless, the number of resources scales with the size of this segment. The more data you want to observe per iteration, the more memory is necessary to process the data at the same time. Figure 4 provides an overview of the architecture we implemented in the case study described in section 5. Although we have followed the approach of the Lambda architecture, the implementation can easily be transfered into a Kappa architecture by removing the corresponding components. In addition, the service layer has to be adjusted as well, since the merging of the two views is omitted. Ultimately, the choice of architecture strongly depends on the respective application and the type of data, necessitating a compromise between consistency, availability and partition tolerance.

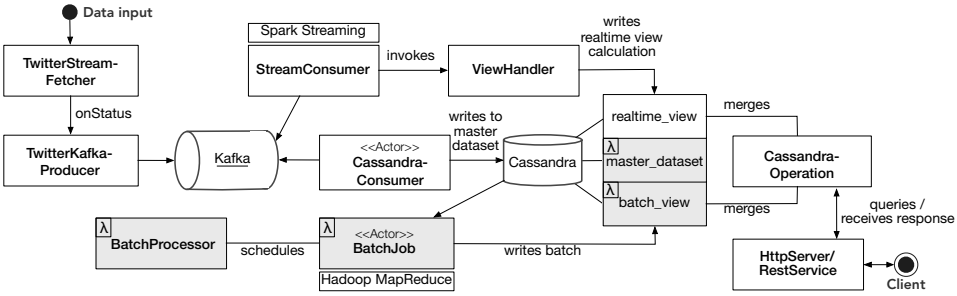


Fig. 4: Overview of the architecture, highlighting components only used in lambda architecture.

7 Conclusion & Future Work

In this paper, we have investigated the state-of-the-art real-time data processing architectures Lambda and Kappa. We started off by taking a closer look at each architecture's components, workflow and theoretical capabilities. While the Lambda architecture was capable of raising awareness about how challenging the development of a shared data system can be, its high complexity remains a considerable disadvantage. Even though the Kappa architecture improves this aspect, the architecture can only be applied for specific use case and might suffer from performance issues. We gave a brief introduction on most commonly known technologies for implementing each architecture's layers as well as the concept of MapReduce. Furthermore, we have discussed the architectures most significant differences that need to be considered when developing a shared data system. After our theoretical investigation, we used Twitter's streaming API for conducting a case study that allows us to gain more detailed insights regarding each architecture's strengths and weaknesses.
We discussed that measuring an architecture's performance with respect to a given use case might not provide particularly sensible information as the result depends on numerous factors. Consequently, future work should focus on prodiving an analysis regarding these factors for allowing an easier decision-making regarding the choice of an architecture.

8 Acknowledgments

We gladly thank Prof. Dr. Markus Esch for his continuous support during the project.

References

[Ab10] Abouzied, Azza et al.: HadoopDB in Action: Building Real World Applications. In: SIGMOD International Conference on Management of data. pp. 1111–1114, 2010.

[Br00] Brewer, Eric A.: Towards Robust Distributed Systems. PODC '00, ACM, New York, NY, USA, pp. 7–, 2000.

[De08] Dean, Jeffrey & Ghemawat, Sanjay: MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1):107–113, 2008.

[Du16] Dunning, T. & Friedman, E.: Streaming Architecture: New Designs Using Apache Kafka and MapR Streams. O'Reilly Media, 2016.

[Ga15] Gandomi, Amir & Haider, Murtaza: Beyond the hype: Big data concepts, methods, and analytics. International Journal of Information Management, 35(2):137–144, 2015.

[Gu15] Gunarathne, T.: Hadoop MapReduce v2 Cookbook - Second Edition. Community Experience Distilled. Packt Publishing, 2015.

[Gu16] Gudipati, Pramod Kumar: Implementing a lambda architecture to perform real-time updates. Master's thesis, Department of Computing and Information Science, Kansas State University, Manhattan, Kansas, 2016.

[Ha14a] Hasani, Zirije et al.: Lambda architecture for real time big data analytic. ICT Innovations, 2014.

[Ha14b] Hasani, Zirije et al.: Survey of technologies for real time big data streams analytic. In: Informatics and Information Technologies. pp. 11–13, 2014.

[Ho12] Holmes, Alex: Hadoop in practice. Manning Publications Co., 2012.

[Ki15] Kiran, Mariam et al.: Lambda architecture for cost-effective batch and speed big data processing. In: Big Data. IEEE, pp. 2785–2792, 2015.

[Kr14] Kreps, Jay: Questioning the Lambda Architecture. 2014.

[KW17] Karau, H.; Warren, R.: High Performance Spark: Best Practices for Scaling and Optimizing Apache Spark. O'Reilly Media, 2017.

[Li14] Liu, Xiufeng et al.: Survey of real-time processing systems for big data. In: IDEAS. 2014.

[Li17] Lin, Jimmy: The Lambda and the Kappa. IEEE Internet Computing, 21(5):60–66, 2017.

[Ma11] Marz, Nathan: How to beat the CAP theorem. Thoughts from the Red Planet, 2011.

[Ma15] Marz, Nathan & Warren, James: Big Data: Principles and Best Practices of Scalable Realtime Data Systems. Manning Publications Co., Greenwich, CT, USA, 2015.

[MJ17] Misra, Pankaj; John, Tomcy: Data Lake for Enterprises. Packt Publishing, 2017.

[Ne13] Neeraj, N.: Mastering Apache Cassandra. Community experience distilled. Packt Publishing, 2013.

[OA16] Ordonez-Ante, Leandro et al.: Interactive querying and data visualization for abuse detection in social network sites. In: Internet Technology and Secured Transactions. IEEE, 2016.

[Up17] Upadhyaya, B.P.: Programming with Scala: Language Exploration. Undergraduate Topics in Computer Science. Springer International Publishing, 2017.

[Zh17] Zhelev, Svetoslav & Rozeva, Anna: Big data processing in the cloud-Challenges and platforms. In: AIP Conference Proceedings. AIP Publishing, 2017.

Anpassung von Stencil-Codes zur Laufzeit mit Hilfe von Umschreiben auf Binärebene für dynamisch bestimmtes Speicherlayout

Konrad M. Pröll[1]

Abstract: Die Optimierung von Stencil-Codes ist eine zentrale Herausforderung im Bereich des Hochleistungsrechnen. Die meisten Ansätze fokussieren sich entweder darauf, diese zur möglichst effizient zur Berechnung auf großen Parallelrechenstrukturen zu parallelisieren oder die möglichst effiziente Ausnutzung des Caches, um die Leistung zu steigern. Viele Stencil-Codes nutzen einfache Arrays zur Speicherung der Matrix. Komplexere Datenstrukturen erhöhen die Rechenzeit des Stencil-Codes dadurch, dass die Berechnung der Speicheradresse einer Zelle deutlich komplizierter wird und sogar bedingte Sprünge enthält. In dieser Arbeit wird ein Ansatz vorgeschlagen, wie aus bereits kompilierten Stencil-Codes zur Laufzeit das Speicherlayout analysiert werden kann und das Programm durch partielle Evaluation optimiert wird. Im Gegensatz zu konventioneller partieller Evaluation wird hierbei nicht für konstante Argumente, sondern für Wertebereiche, in denen sich ein Argument befindet, spezialisiert. Durch diese Methode können die Leistungseinbußen merklich reduziert werden.

Keywords: Programmoptimierung; Partielle Evaluation; Stencil-Codes; High Performance Computing; Umschreiben auf Binärebene

1 Einleitung

Stencil-Codes sind eine Art von Programmen, die häufig im Bereich des wissenschaftlichen Rechnens zum Einsatz kommen. Hierbei wird der Wert sämtlicher Zellen einer Matrix unter Berücksichtigung der Werte der Nachbarzellen in jeder Iteration neu berechnet, indem ein gewichteter Durschnitt dieser gebildet wird. Die Form des Stencil bestimmt, welche Nachbarzellen bei der Neuberechnung berücksichtigt werden. Stencil-Codes können für alle naturwissenschaftlichen Probleme, die auf das Lösen von Differenzialgleichungen reduzierbar sind, verwendet werden. Diese Gruppe von Problemen beinhaltet physikalische Simulationen (Strömungsdynamik, Ausbreitung von Wärme, Auswirkungen von Erdbeben), aber auch Bereiche der Quantenmechanik oder der Bildverarbeitung [Da09; RYQ11]. Sie werden genutzt, um kontinuierliche Differenzialgleichungen zu diskretisieren, sodass diese überhaupt erst berechenbar werden.

Häufig werden Stencil-Codes für sehr große Matrizen über eine große Anzahl an Iterationen verwendet, was zu einer hohen Rechenzeit führt, weshalb die Optimierung dieser Programme

[1] Technische Universität München, Fakultät für Informatik, konrad.proell@tum.de

ein wichtiges Forschungsgebiet im Bereich des Hochleistungsrechnen (HPC) ist. Die meisten Optimierungen fokussieren sich auf das Anpassen des Programms zur Ausführung auf Parallelarchitekturen, z. B. über Message Passing Interface (MPI), nicht zuletzt weil die einzelnen Neuberechnungen innerhalb eines Iterationsschrittes meist nur von den Werten des vorhergehenden Schrittes abhängt, sodass das Parallelisieren grundsätzlich ohne größere Schwierigkeiten möglich ist, sowie zur effektiven Ausnutzung des Caches, da einfache Stencil-Codes meist durch die Speicherbandbreite der Maschine limitiert werden [Kr07].

Gewöhnlich werden die Matrizen für Stencil-Codes in normalen Arrays gespeichert, sodass die Speicheradressen der Zellen mittels Offset-Berechnung ermittelt werden können. Unter Umständen kann es vorteilhaft sein, ein dynamisch bestimmtes Speicherlayout zu verwenden und verschiedene Teile der Matrix an verschiedenen Adressen abzulegen. Solche dynamisch bestimmte Speicherlayouts besitzen den Nachteil, dass die Berechnung der Speicheradresse einer Zelle deutlich komplizierter ist und sogar bedingte Sprünge enthält.

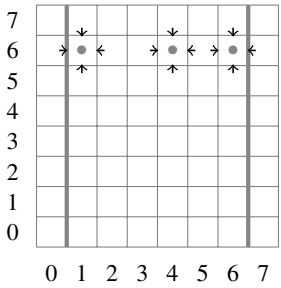


Abb. 1: Beispiel für ein Speicherlayout. Die linke und rechte äußere Spalte werden an anderen Orten als der Rest der Matrix gespeichert, etwa um die “Halo”-Zellen bei Parallelisierung dichter abzuspeichern.

In Abb. 1 ist eine Matrix zu sehen, die in drei verschiedenen Arrays abgelegt ist: Eines für die äußere linke Spalte, eines für die äußere rechte sowie eines für den Rest. Ein solches Speicherlayout kann z. B. sinnvoll sein, um die “Halo”-Zellen, die bei einer Parallelisierung auf Architekturen mit verteiltem Speicher genutzt werden [KS10], an einer anderen Stelle im Speicher abzulegen, sodass sie dichter im Speicher liegen. Im Falle eines solchen Speicherlayouts muss bei jeder Ausführung des Stencil-Kernels (d. h. bei jeder Neuberechnung eines Wertes) für jeden Punkt des Stencils geprüft werden, in welchem Array er liegt. Ein gewöhnlicher 4-Punkt-Stencil enthält also bei dem dargestellten Layout 4 - 8 bedingte Sprünge, die die Rechnzeit eines Stencil-Codes erheblich erhöhen.

In diesem Paper wird eine Methode vorgestellt, um zuerst aus dem kompilierten Stencil-Kernel das Speicherlayout in einer Analysephase zu rekonstruieren (die möglichen Varianten des Kernels ermitteln), und den Kernel mittels partieller Evaluation für dieses Speicherlayout zu optimieren. Die weitere Arbeit ist folgendermaßen aufgebaut: Zuerst werden die bearbeitete Problemstellung sowie verwandte Arbeiten vorgestellt. Daraufhin soll das Projekt, in das die Methode eingebettet wird, knapp vorgestellt werden, bevor die konkrete

Vorgehensweise präsentiert wird. Anschließend wird die Implementierung in Bezug auf die Verbesserung der Laufzeit evaluiert, bevor die Ergebnisse beurteilt werden.

Die Forschungsbeiträge dieser Arbeit sind:

- Identifizieren des Speicherlayouts aus kompiliertem Stencil-Kernel
- Optimieren von Stencil-Kernels für dynamisch bestimmtes Speicherlayout mittels partieller Evaluation für Wertebereiche

2 Verwandte Arbeiten

In diesem Abschnitt sollen verwandte Arbeiten vorgestellt und diskutiert werden.

Spezialisierung bzw. partielle Evaluation ist eine Programmtransformationstechnik, bei der vor der eigentlichen Programmausführung Teile bereits ausgewertet werden, die von Variablen abhängig sind, die bei jedem Aufruf gleich sind. Hierbei wird das Programm auf jene Variablen "spezialisiert". Spezialisierung wird vor allem aus zwei Gründen eingesetzt: Zum einem zur Generierung effizienter Compiler[2][JGS93, S. 13 ff.], und zum anderen zur Optimierung von Programmen.

Srinivasan und Reps [SR15] stellen ein Konzept zur Spezialisierung von Maschinencode vor. Dabei wird der Maschinencode in einen Syntaxbaum umgewandelt, in dem sowohl die Abhängigkeiten der einzelnen Prozeduren untereinander als auch die Abhängigkeiten innerhalb dieser analysiert werden. Im Anschluss können mittels "Slicing" die Abhängigkeiten der einzelnen Befehle untereinander analysiert werden. Auf dieser Ebene kann dann der Programmcode für verschiedene konstante Parameter spezialisiert werden. In [SR16] stellen die Autoren eine verbesserte Methode des Slicing vor.

Köster et al. [Kö14] verwenden Spezialisierung für Stencil-Codes, um Programme, die für ein variables Stencil-Layout entwickelt wurden, auf ein konkretes zu spezialisieren. Im Rahmen ihrer Arbeit haben sie eine Zwischenrepräsentation entwickelt, die Programme in einer Baumstruktur speichert und auf jegliche Blockstrukturen verzichtet. Jeder Knoten des Baums stellt eine Definition dar, während Kanten für direkte Abhängigkeiten zwischen Definitionen stehen. Auf dieser Ebene wird dann der Programmcode für bestimmte Variablen spezialisiert und alle von diesem Parameter abhängige Ausdrücke ausgewertet [LKH15]. Die Autoren messen bei der Verwendung eines Jacobi-Kernel spürbar bessere Leistungen für die spezialisierte Variante gegenüber der variablen bei einer Beschleunigung um einen Faktor zwischen 2 und 3.

Diese Ansätze teilen mit anderen geläufigen Konzepten der Spezialisierung, dass hierbei der Programmcode für *konstante* Parameter spezialisiert wird. Im Fall eines dynamisch

[2] Durch Spezialisierung eines Spezialisierers für einen Interpreter wird ein Compiler generiert

bestimmten Speicherlayouts soll nicht für einen konstanten Wert, sondern einen Wertebereich spezialisiert werden. Daher wird ein neuer Ansatz benötigt, in dem Funktionen nicht nur für konstante Werte, sondern ganze Wertebereiche spezialisiert werden.

3 Hintergrund

Am Lehrstuhl für Rechnerarchitektur & Parallele Systeme der Technischen Universität München[3] wird seit 2015 im Projekt **DBrew**[4] eine C-Bibliothek entwickelt, die es ermöglicht bereits kompilierten Code mittels Umschreiben auf Binärebene zu optimieren. Hierbei wird der Binärcode in Maschinencode, strukturiert in Basic Blocks, dekodiert und nach der Optimierung wieder in Binärcode konvertiert.

Im ersten Schritt kann der dekodierte Programmcode spezialisiert werden. Hierbei kann der Programmierer diejenigen Parameter, die spezialisiert werden, als "statisch" setzen. Anschließend wird ein Funktionsaufruf emuliert. Hierbei wird für alle Werte gespeichert, ob sie statisch sind. Operationen, deren Ergebnisse statisch sind, werden hierbei bereits während der Emulation ausgeführt. Basic Blocks werden während der Emulation in einem Stack-Speicher verwaltet: Zu Beginn wird derjenige Block, der die Sprungmarke der jeweiligen Funktion enthält, auf dem Stack abgelegt und bearbeitet, während in den weiteren Schritten immer derjenige Block auf dem Stack abgelegt wird, der auf den gerade bearbeiteten Block folgt. Falls ein Block mit einem bedingten Sprung endet, sodass mehrere verschiedene Blöcke auf ihn folgen können, wird zuerst geprüft, ob die Sprungbedingung aufgrund der Parameter, für die spezialisiert wird, ausgewertet werden kann. Sollte dies der Fall sein, wird der Sprungbefehl durch einen unbedingten Sprung ersetzt, ansonsten werden beide möglichen Blocks auf dem Stack abgelegt [WB16].

Im Anschluss an die Emulation kann der Maschinencode mittels LLVM weiter optimiert werden [EW17]. Abschließend wird ein Pointer zurückgegeben, über den die Funktion aufgerufen werden kann.

4 Vorgehensweise

Im Folgenden soll die konkrete Vorgehensweise vorgestellt werden.

4.1 Metainformationen

Um Wertebereiche für Eingabeparameter festzustellen, für die Sprungbedingungen andere Werte ergeben, wird zuerst für jeden Wert (d. h. Register, Flag und Stack) ein **symbolischer**

[3] `https://www.lrr.in.tum.de/startseite/`
[4] Dynamic Binary rewriting

Ausdruck gespeichert, der die Abhängigkeit des Wertes von den Parametern der Funktion beschreibt. Symbolische Ausdrücke werden in einem Ausdrucksbaum gespeichert: Die Blätter beinhalten entweder einen konstanten Wert, falls der jeweilige Wert von keinem Parameter abhängt, oder den Parameter, von dem der jeweilige Wert abhängt. Die Knoten im Baum werden genutzt, um mathematische Operationen zu speichern. In Abb. 3 ist ein solcher Ausdrucksbaum abgebildet, der die Operation $(x + 5) * 2$ speichert, wobei x ein Parameter der Funktion ist.

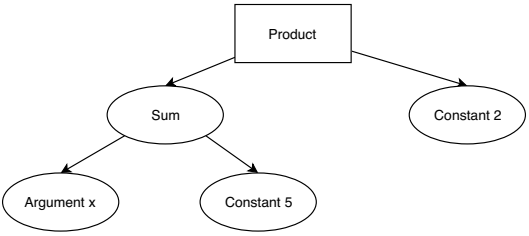


Abb. 2: Aufbau des symbolischen Ausdrucks für $(x + 5) * 2$

Im nächsten Schritt wird für jeden Wert ein **Wertebereich** eingeführt, der jeweils den minimalen und maximalen Wert, den ein Ergebnis annehmen kann, beschreibt. Eine Variable kann auch mehrere disjunkte Wertebereiche haben. Anfangs setzt der Nutzer den Wertebereich der Parameter der Funktion. Anschließend wird bei jeder Operation der Wertebereich des jeweiligen Ergebnisses berechnet.

Abschließend wird in einem **Kontrollflussgraph** die Struktur des Programmes festgehalten. In jedem Knoten wird hierbei ein Basic Block gespeichert, während Kanten zeigen, welche Knoten von einem Basic Block aus erreicht werden können. Zusätzlich wird die jeweils letzte Instruktion des Blocks gespeichert, da diese den Kontrollfluss betrifft, sowie etwaige Sprungbedingungen in Form eines symbolischen Ausdrucks und die Wertebereiche aller Funktionsargumente, für die der jeweilige Knoten erreicht werden kann.

4.2 Analyse

Nachdem die Metainformationen, die während eines Durchlaufs zusätzlich betrachtet werden, vorgestellt wurden, soll nun gezeigt werden, wie mit diesen mögliche Spezialisierungen identifiziert werden können.

Anfangs setzt der Nutzer die Wertebereiche der Funktionsparameter auf die zu erwartbaren Werte. Im Anschluss wird das ein Funktionsaufruf wie in Abschnitt 3 beschrieben emuliert. Hierbei wird denjenigen Registern, in denen die Funktionsargumente stehen, symbolische Ausdrücke angehängt, dass das Parameter der Funktion sind.

Im weiteren Verlauf der Emulation wird zwischen Instruktionen, die den Kontrollfluss des Programms nicht beeinflussen (einfache Instruktionen), und solchen, die den Kontrollfluss beeinflussen unterschieden.

Einfache Instruktionen werden folgendermaßen bearbeitet:

1. Alle Operanden der Instruktion sind konstant: In diesem Fall hängt das Ergebnis nicht (mehr) von Argumenten der Funktion ab. Deshalb kann eine solche Instruktion einfach emuliert werden, der symbolische Ausdruck des Ergebnisses wird auf eine Konstante gesetzt.
2. Es gibt nicht-konstante Argumente, aber für jeden dynamischen Operanden liegt ein symbolischer Ausdruck vor: Die Instruktion wird symbolisch ausgeführt, sodass ein neuer Ausdruck entsteht, der das Ergebnis der Operation als Ausdrucksbaum enthält. Anschließend wird der Wertebereich des Ergebnisses ermittelt. Sollte der Wertebereich nur einen Wert umfassen, ist das Ergebnis konstant, sodass der ermittelte Ausdruck verworfen und wiederum durch einen symbolischen Ausdruck für eine Konstante ersetzt werden kann, andernfalls sind der zuvor ermittelte Ausdruck und Wertebereich das Ergebnis dieser Operation.
3. Es gibt nicht-konstante Operanden, für die kein symbolischer Ausdruck vorliegt[5]: In diesem Fall können Abhängigkeiten nicht beurteilt werden, weshalb auch für das Ergebnis kein Ausdruck ermittelt werden kann. Wenn dieses Ergebnis in weiteren Instruktionen referenziert wird, kann auch über deren Ergebnis keine Aussage getroffen werden.

Instruktionen, die den **Kontrollfluss** beeinflussen, können zu unterschiedlichen Spezialisierungen führen, weswegen sie gesondert behandelt werden müssen. Solche Instruktionen können, müssen aber nicht zwingend bedingte Sprünge sein. Beispiele für weitere solche Instruktionen sind *ret* oder *jmp*. Sie werden folgendermaßen bearbeitet:

1. Der Sprung wird ohne Bedingung ausgeführt oder die Sprungbedingung ist konstant: Diese Art von Instruktion beeinflusst die möglichen Spezialisierungen nicht. Im Fall von Sprungbefehlen wird der Kontrollflussgraph nicht aktualisiert und der nächste Block wird zum aktuellen Knoten hinzugefügt. Im Falle eines *ret* wird diese Instruktion im Knoten gespeichert, da die Funktion an dieser Stelle des Kontrollflussgraphen beendet ist.
2. Teile der Sprungbedingungen sind dynamisch, und es liegen für alle dynamischen Sprungbedingungen symbolische Ausdrücke vor: Dies sind diejenigen bedingten Sprünge, für die das Programm anschließend spezialisiert werden soll. Die Sprungbedingung wird symbolisch ausgewertet, sodass ein neuer symbolischer Ausdruck entsteht, der beschreibt, wann die Bedingung erfüllt ist. Sollte der Ausdruck nur eine Variable (d. h. Funktionsparameter) enthalten, wird er danach aufgelöst[6]. Das Ergebnis

[5] z. B. weil die Verarbeitung der Instruktion noch nicht implementiert wurde oder die Variable global außerhalb der Funktion definiert wurde

[6] Das Auflösen komplexerer Ausdrücke führt zu sehr feingranularen Spezialisierungen. Funktionen für Wertebereiche zu spezialisieren, die ohnehin kaum genutzt werden, verspricht wenig Leistungsverbesserung und blockiert unverhältnismäßig Speicher.

dieser Operation wird neben dem genauen Sprungbefehl im aktuellen Knoten gespeichert. Anschließend werden an diesen für die beiden möglichen Ausführungspfade zwei Kinder angehängt. In beide Kinder wird auf Basis der Sprungbedingung und der Wertebereiche im Elternknoten derjenige Wertebereich der Funktionsparameter berechnet.

3. Dynamische Sprungbedingungen, für die keine symbolischen Ausdrücke vorliegen: Diese Sprungbedingungen können nicht analysiert werden. An den Knoten des aktuellen Blocks werden zwei Kinder angehängt, in die die Wertebereiche der Funktionsargumente des aktuellen Knotens kopiert wird. Dadurch können trotzdem noch folgende Sprünge analysiert werden, sollten sie verwertbare Sprungbedingungen haben.

Am Ende des Analyselaufs können die Ergebnisse dem Kontrollflussgraphen entnommen werden. Wie in Abb. 3 zu sehen ist, enthalten sämtliche Knoten jeweils, für welche Wertebereiche der Funktionsparameter diese erreicht werden kann, die letzte Instruktion und, falls es sich hierbei um einen bedingten Sprung handelt, die Sprungbedingung. In der Beispielfunktion, die einen Parameter hat, wurde der Wertebereich des Arguments auf $[0, 9]$ festgesetzt. In der Folge trifft die Programmausführung auf einen bedingten Sprungbefehl *JL*[7], wobei die Sprungbedingung ein Vergleich des Funktionsarguments mit 5 ist. Die linke Hälfte des Baums wird deshalb erreicht, wenn das Argument < 5 ist, weswegen im entsprechenden Knoten als Intervall $[0, 4]$ gespeichert ist. Am Ende können die Spezialisierungen den Blättern entnommen werden: Das linke Blatt kann nicht erreicht werden, weswegen kein Wertebereich darin gespeichert ist. Das zweite Blatt wird für $X \in [0, 4]$ erreicht, das dritte für $X = 5$ und das letzte für $X \in [6, 9]$. Deshalb können durch Spezialisierung dieser Varianten alle bedingten Sprünge entfernt werden.

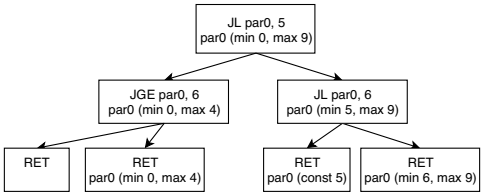


Abb. 3: Beispiel eines Kontrollflussgraphen am Ende eines Analyselaufs. Es wurden drei mögliche Spezialisierungen festgestellt, die den Blättern des Baumes entnommen werden: $X\ in [0, 4]$, $X = 5$ sowie $X \in [6, 9]$.

4.3 Spezialisierung

Nachdem die gegebene Funktion analysiert worden ist und dadurch diejenigen Wertebereiche identifiziert wurden, für die die Sprungbedingungen der bedingten Sprünge zum gleichen

[7] **J**ump if **L**ess, wird ausgeführt, wenn der erste Operand kleiner als der zweite ist (d. h. $SF \neq OF$ [In16])

Ergebnis führen, kann die Funktion durch weitere Läufe von DBrew für die einzelnen Bereiche spezialisiert werden. Hierbei werden zu Beginn die Wertebereiche der Parameter auf die in der Analyse festgestellten gesetzt und eine weitere Emulation gestartet. Die Spezialisierung wird hierbei dadurch erwirkt, dass durch die engeren Wertebereiche Sprungbedingungen, die während der Analyse noch dynamisch waren, konstant werden und ausgewertet werden können, sodass die bedingten Sprungbefehle durch unbedingte Sprünge ersetzt werden.

5 Evaluation und Diskussion

In diesem Abschnitt soll die zuvor vorgestellte Methode zur Spezialisierung evaluiert werden. Hierbei wird zuerst das benutzte System vorgestellt, bevor die erzielten Geschwindigkeitszunahmen in verschiedenen Testfällen gezeigt werden. Abschließend werden die Ergebnisse diskutiert.

5.1 Testaufbau

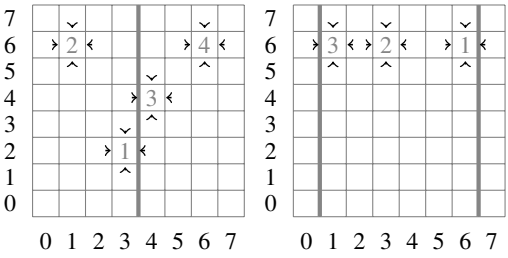
Als Testsystem wurde ein Intel ®Core ™i3-2310m mit 2.1 GHz mit dem Betriebssystem Ubuntu 17.10 verwendet. Als Compiler wurde *gcc 7.2.0* mit den Optionen *-O2 -mavx* genutzt. Die genutzte Kernelversion ist *4.13.0-36-generic*. In jedem Durchlauf wurde der Stencil-Code über 50 Iterationen bei einer zweidimensionalen Matrix mit der Auflösung 2000*2000 ausgeführt. Jede dieser Messungen wurde 200 mal wiederholt. Sämtliche Messungen wurden seriell durchgeführt.

5.2 Messergebnisse

Im Folgenden soll anhand zweier Testfälle der Funktionsumfang der Analyse- und Spezialisierungsphase anhand eines Jacobi-Kernels, der zur Approximation der Wärmeleitungsgleichung verwendet wird, gezeigt werden. Die drei Testfälle sind in Abb. 4 dargestellt. Laufzeitmessungen werden für fünf Konfigurationen vorgenommen:

- Standard: In dieser Konfiguration wird die Domäne entlang der Zeilen durchlaufen.
- Sortiert: In dieser Konfiguration wird die Ausführreihenfolge so angepasst, dass innerhalb einer Iteration alle Aufrufe, die zu einem Speicherlayout gehören, durchlaufen werden, bevor die Ausführung zum nächsten Layout übergeht. Außerdem werden innerhalb eines Bereichs die Schleifen so angepasst, dass der Cache möglichst gut ausgenutzt wird. Hierdurch soll außerdem getestet werden, ob ein moderner Branch-Predictor einen ähnlichen Effekt wie die Spezialisierung erzielen kann.

- Matrix: In dieser Konfiguration wird DBrew genutzt, um den Stencil-Kernel mittels gewöhnlicher Spezialisierung für die Größe der Matrix spezialisiert.
- Matrix, sortiert: In dieser Konfiguration wird zuerst der Kernel wie in der vorherigen Variante spezialisiert. Anschließend werden die Schleifen wie in der Variante zuvor sortiert.
- Spezialisiert: In dieser Variante wird zuerst der Stencil-Code wie in dieser Arbeit vorgeschlagen spezialisiert. Die Ausführreihenfolge ist identisch zur Option “sortiert”.



(a) Testfall 1 (b) Testfall 2

Abb. 4: Behandelte Testfälle. In Abb. 5a wird die Matrix in der Mitte halbiert, weswegen vier verschiedene Spezialisierungen möglich sind. In Abb. 5b werden die linke und rechte Spalte der Matrix an einer anderen Adresse gespeichert, etwa um *Halo*-Zellen bei einer Parallelisierung dichter abzulegen. Die nummerierten Felder stellen die unterschiedlichen Spezialisierungen dar.

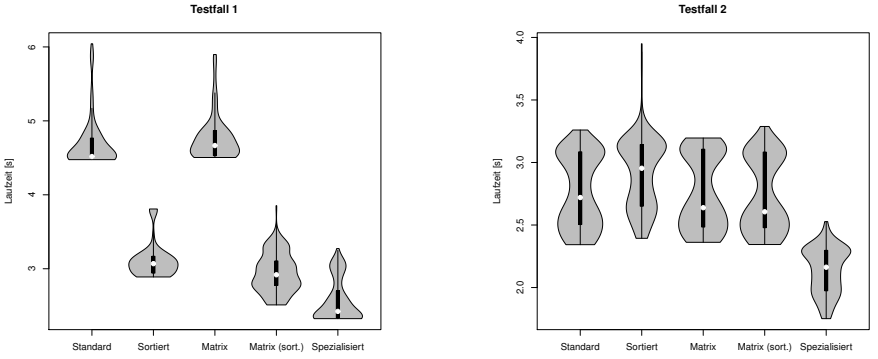
Testfall 1

In Testfall 1 wird die Matrix in der Mitte vertikal in zwei Hälften geteilt. Die Analysephase erkennt korrekt die vier möglichen Spezialisierungen $X = 3$, $X \in [1, 2]$, $X = 4$ und $X \in [5, 6]$. Hierbei wird die linke Hälfte zeilenorientiert (“row-major”) und die rechte Hälfte spaltenorientiert (“column-major”) gespeichert.

Wie in Abb. 5 zu sehen ist, wird durch das Umsortieren ein erheblicher Geschwindigkeitsvorteil erzielt, da dann auch bei der spaltenorientierten rechten Hälfte der Cache gut ausgenutzt wird. Durch das Spezialisieren für die einzelnen Speicherbereiche wird ein Geschwindigkeitszuwachs von etwa 8% im Vergleich zur gewöhnlichen Spezialisierung erzielt.

Testfall 2

In Testfall 2 wird die Matrix in drei Teile geteilt, wobei jeweils die äußerste linke und rechte Spalte an einer anderen Adresse gespeichert werden. Dies kann z. B. dazu genutzt werden,



(a) Testfall 1

(b) Testfall 2

Abb. 5: Ergebnisse der Laufzeitmessungen. In beiden Testfällen kann durch Nutzung spezialisierter Kernel für die jeweiligen Ausführungspfade die Laufzeit verringert werden.

"Halo"-Zellen beim Parallelisieren für eine Architektur mit verteiltem Speicher an einer anderen Stelle im Speicher dichter abzulegen. Die Analysephase erkennt korrekt die drei möglichen Spezialisierungen $X = 6$, $X \in [2, 5]$ und $X = 1$.

Wie Abb. 5 zu entnehmen ist, ist in diesem Fall die Spezialisierung etwa 10% schneller. Anders als im vorherigen Fall führt das Umsortieren der Schleifen hier zu einer Verlangsamung, weil dadurch das Cache-Verhalten nicht verbessert, sondern verschlechtert wird. Der höhere Speedup der Spezialisierung als im vorherigen Fall erklärt sich dadurch, dass in diesem Fall für jeden Punkt zwei Bedingungen statt nur einer geprüft werden müssen.

5.3 Diskussion

Der vorgestellte Ansatz führte zu einer merklichen Leistungsverbesserung: Speedups im Bereich von 8% - 10% sind zwar im Vergleich zum Potential in Optimierungen zur

besseren Cachenutzung recht gering, sind aber unter Berücksichtigung dessen, dass einfache Stencil-Codes stark durch die Speicherbandbreite limitiert werden, durchaus bemerkbar.

Ein Nachteil dieses Ansatzes, der bei den untersuchten Speicherlayouts nicht auftritt, ist, dass unter Umständen toter Code generiert wird: Wenn Sprungbedingungen von komplexen Berechnungen abhängen, und die Sprungbedingung in der Spezialisierung konstant wird, trifft dies nicht zwingend auf die vorherigen Berechnungen zu. Deshalb werden dann bei jedem Durchlauf diese Berechnungen durchgeführt, aber das Ergebnis nicht weiter genutzt. Durch die anschließende Transformation des spezialisierten Codes in eine Zwischenrepräsentation wie LLVM kann dieses Problem behoben werden.

Eine weitere Hürde ist, dass komplexere Speicherlayouts zu sehr vielen Spezialisierungen führen kann. In diesem Fall könnte z. B. über eine Analyse festgestellt werden, welche Varianten besonders oft ausgeführt werden, und dann nur diese spezialisieren und für die restliche Ausführung die generische Funktion nutzen.

6 Fazit

Im Rahmen dieser Arbeit wurde vorgeschlagen, wie Stencil-Kernel für dynamisch bestimmte Speicherlayouts mittels Spezialisierung für Wertebereiche optimiert werden. Zuerst wurde gezeigt, wie im Rahmen einer Analysephase das Layout so identifiziert werden kann, dass die Wertebereiche der Eingabeparameter, für die die einzelnen Punkte des Stencils auf den gleichen Speicherbereich zugreifen, festgestellt werden. Anschließend wurde der Kernel so spezialisiert, dass die Teile des Programms, die für alle Werte in diesem Bereich konstant sind, bereits ausgewertet wurden, während diejenigen, die vom tatsächlichen Wert abhängen, erst zur Laufzeit berechnet werden. Es wurde gezeigt, dass dadurch Beschleunigungen im Bereich von 10% erzielt werden können.

Danksagung

Diese Arbeit ist im Rahmen meiner Masterarbeit am Lehrstuhl für Rechnerarchitektur & Parallele Systeme entstanden. Mein Dank gilt Josef Weidendorfer von Alexis Engelke für die Betreuung dieser Arbeit.

Literatur

[Da09] Datta, K.: Auto-tuning Stencil Codes for Cache-based Multicore Platforms, AAI3411221, Diss., Berkeley, CA, USA, 2009, ISBN: 978-1-124-03708-0.

[EW17] Engelke, A.; Weidendorfer, J.: Using LLVM for Optimized Lightweight Binary Re-Writing at Runtime. In: 2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). S. 785–794, Mai 2017.

[In16] Intel Corporation: Intel® 64 and IA-32 Architectures Software Developer's Manual Volume 2A: Instruction Set Reference, A-L, 253666-060US, Intel Corporation, Sep. 2016.

[JGS93] Jones, N. D.; Gomard, C. K.; Sestoft, P.: Partial Evaluation and Automatic Program Generation. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[Kö14] Köster, M.; Leißa, R.; Hack, S.; Membarth, R.; Slusallek, P.: Platform-Specific Optimization and Mapping of Stencil Codes through Refinement. In: Proceedings of the First International Workshop on High-Performance Stencil Computations (HiStencils). Vienna, Austria, S. 1–6, 21. Jan. 2014.

[Kr07] Krishnamoorthy, S.; Baskaran, M.; Bondhugula, U.; Ramanujam, J.; Rountev, A.; Sadayappan, P.: Effective Automatic Parallelization of Stencil Computations. In: Proceedings of the 28th ACM SIGPLAN Conference on Programming Language Design and Implementation. PLDI '07, ACM, San Diego, California, USA, S. 235–244, 2007, ISBN: 978-1-59593-633-2.

[KS10] Kjolstad, F. B.; Snir, M.: Ghost Cell Pattern. In: Proceedings of the 2010 Workshop on Parallel Programming Patterns. ParaPLoP '10, ACM, Carefree, Arizona, USA, 4:1–4:9, 2010, ISBN: 978-1-4503-0127-5.

[LKH15] Leißa, R.; Köster, M.; Hack, S.: A Graph-based Higher-order Intermediate Representation. In: Proceedings of the 13th Annual IEEE/ACM International Symposium on Code Generation and Optimization. CGO '15, IEEE Computer Society, San Francisco, California, S. 202–212, 2015, ISBN: 978-1-4799-8161-8.

[RYQ11] Rahman, S. M. F.; Yi, Q.; Qasem, A.: Understanding Stencil Code Performance on Multicore Architectures. In: Proceedings of the 8th ACM International Conference on Computing Frontiers. CF '11, ACM, Ischia, Italy, 30:1–30:10, 2011, ISBN: 978-1-4503-0698-0.

[SR15] Srinivasan, V.; Reps, T.: Partial Evaluation of Machine Code. In: Proceedings of the 2015 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications. OOPSLA 2015, ACM, Pittsburgh, PA, USA, S. 860–879, 2015, ISBN: 978-1-4503-3689-5.

[SR16] Srinivasan, V.; Reps, T.: An Improved Algorithm for Slicing Machine Code. In: Proceedings of the 2016 ACM SIGPLAN International Conference on Object-Oriented Programming, Systems, Languages, and Applications. OOPSLA 2016, ACM, Amsterdam, Netherlands, S. 378–393, 2016, ISBN: 978-1-4503-4444-9.

[WB16] Weidendorfer, J.; Breitbart, J.: The Case for Binary Rewriting at Runtime for Efficient Implementation of High-Level Programming Models in HPC. In: 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). S. 376–385, Mai 2016.

Business IT

Success Factors in Business-Managed IT - A Case Study Analysis at a Large German Industrial Company

Matthias Bachfischer[1]

Abstract: The terminology of Business-Managed IT refers to Shadow-IT systems which are operated overtly in the business units (BUs) and with the awareness from the IT department. They are a common phenomenon in corporations and usually emerge if the formal IT organization is unable to provide the BUs with solutions that meet their requirements. Because of this, Business-Managed IT is a highly significant area for an exploratory study, and both academia and practitioners can benefit from knowledge on how such systems can be successfully managed. In this present research, a case study analysis of five Business-Managed IT systems at a large German industrial company was conducted. Interviews with stakeholders involved in these systems were transcribed and analyzed to derive Critical Success Factors (CSFs) for Business-Managed IT. After a careful analysis of the data, a total of nine CSFs for Business-Managed IT systems were obtained. They fall into three dimensions: *Project Dimension, Organizational Dimension* and *System Dimension*.

Keywords: Business-Managed IT; Shadow-IT; IT-Organization; Critical Success Factors

1 Introduction

In recent years, many traditional Information Technology (IT) departments in corporations have been struggling with their role as an enabler for successful business activities by providing high quality IT services in a dynamic market environment [UA16]. Reasons for this are, amongst others, long project cycle times for too large and too complex IT projects as well as poor alignment between the business and IT departments within a corporation.
Many business units (BUs) have therefore started to build and develop their own "systems, services, and processes that are not part of the official corporate IT" [KW16a]. In some corporations, BUs even started to create their own (informal) IT organizations who are responsible for maintaining systems that are "autonomously established by [the] business units" [FR14]. Systems such as these which are decentrally and covertly operated by BUs and not part of the official corporate IT are generally referred to as Shadow-IT.
Until recently, academic research in the area of Shadow-IT has mostly been focusing on the negative aspects of Shadow-IT and its "lack of integration in the enterprise architecture" [Hu16]. Shadow-IT is furthermore seen as an "IT risk for organisational information security" [SB14] and considered to be of *covert* nature because "related activities are

[1] Ostbayerische Technische Hochschule Regensburg, Faculty of Computer Sciences and Mathematics, bachfischer.matthias@googlemail.com

practiced in a hidden form" [Ko18]. Contrary to this, the concept of Business-Managed IT refers to the phenomenon of *overt* Information Systems (ISs) where "the related activities regarding . . . development and operation [of the IT system] are practiced openly" [Ko18] and are, to a certain extent, aligned with the activities of the IT department.
The intention of this research is to determine which factors contribute to the success of Business-Managed IT. First, the terminology is defined and related literature is reviewed. Next, the research methodology and the context of the case study is presented and following that, the resulting Critical Success Factors (CSFs) obtained from the case study are discussed. The paper is concluded with an outline of the limitations and opportunities for future research.

2 Definitions and Related Works

A comprehensive definition of Shadow-IT was found by Zimmermann, Rentrop and Felden who described it as a "business process supporting IT systems, IT service processes, and IT staff. Shadow-IT is deployed autonomously within business departments and by IT users. Thereby, Shadow-IT systems are involved neither technically nor strategically in the IT service management of the organization" [ZRF14].
However, and contrary to the common belief that Shadow-IT mostly entails negative consequences for the operation of a business, recent publications emphasize the positive consequences of allocating responsibility for IT systems decentrally in the BUs [Ko17]. Academia therefore proposed to differentiate between Shadow-IT and Business-Managed IT and defined Business-Managed IT as the practice of operating ISs within the BUs which are involved in the organizational IT management [Ko17] and operated in alignment with the IT department [Ko18].

Even though the term Business-Managed IT was presumably first introduced by Kopper in 2017, the idea to "reallocate responsibilities and share the tasks of identified Shadow-IT between the business and the IT units" [ZRF16] as well as the trajectory of "Central IT assumes Maintenance [of Shadow-IT systems]" [CSC14] has already been presented in previous works. To understand the existing body of knowledge and explore areas for future research [LE06], the first step of this research was to conduct an extensive literature review following the recommendations of Levy and Ellis (2016). First, the literature databases AISeL[2], EBSCOhost[3], IEEE Xplore[4] and ScienceDirect[5] were queried with a variety of keywords related to the concept of Shadow-IT, such as *Shadow-IT, Business-Managed IT* or *Shadow systems*. Next, and based on the publications yielded from the search in literature databases described above, a backward and forward references search was conducted [LE06]. As a result, 26 publications related to the concept of Shadow-IT were obtained.
The majority of previous research in the field of Shadow-IT has focused on analyzing

[2] AISeL - Website `http://aisel.aisnet.org`
[3] EBSCOhost - Website `https://www.ebsco.com/`
[4] IEEE Xplore `https://ieeexplore.ieee.org`
[5] ScienceDirect - Website `https://www.sciencedirect.com/`

literature to "identify motivators, enablers, and missing barriers as causes for Shadow-IT" [KW16a] as well as to create a "taxonomy of Shadow-IT related concepts" [KW16b]. Less research has focused on the empirical consequences of Shadow-IT and what kind of measures organizations should implement to manage Shadow-IT.
A promising approach was taken in 2017 by Zimmermann, Rentrop and Felden who conducted several case-studies amongst three corporations to study the "nature of Shadow-IT in organizations" and to determine how "organizations [can] manage identified Shadow-IT instances" [ZRF17]. Another recent publication from Kopper presented "exploratory interviews with 16 executive/senior IT managers" [Ko17] and "revealed different perspectives on Shadow-IT and related organizational practices" in organizations [Ko17]. In 2018, Kopper, Fürstenau et al. furthermore carried out a study using four case examples to study the differences between Shadow-IT and Business-Managed IT and proclaimed that Business-Managed IT has a high potential to "balance the tension between speed/autonomy and cost-effectiveness/safety/risk" [Ko18].
This present research used the CSF approach to determine factors that contribute to the success of Business-Managed IT. The CSF method is frequently used within IS research because it allows the researcher to determine CSFs that "are useful in prioritizing potential information systems projects by identifying those information services that address critical organizational concerns" [BB94]. In particular, the study conducted by Chow & Cao (2008) which obtained "five factor categories, namely Organizational, People, Process, Technical, and Project" [CC08] proved to be a highly valuable resource and was used as a reference for deriving the CSF dimensions from the case study results presented in this paper.

3 Research Methodology

3.1 Case Study

This paper performs an exploratory case study analysis on the subject of distinct Business-Managed IT systems at a large industrial company. The research method of a case study is deliberately chosen due to the fact that conducting a case study analysis can be especially useful when the objective is to create an "extensive and 'in-depth' description" [Yi09] of a specific subject. If no established body of knowledge in the research area exists, case studies can furthermore be used for theory building with the purpose of creating "theoretical constructs, propositions and/or midrange theory from case-based, empirical evidence" [EG07].
At the beginning of the case study, the propositions to be examined were developed and the case to be studied was selected. As described in Section 4, the research context of this paper were Business-Managed IT systems at a large industrial company. In order to obtain a better understanding of Business-Managed IT systems at the company, five in-depth interviews with stakeholders who were involved with these systems were conducted. They were recorded using a recording device and transcribed by using the software f4transcript [6]

[6] f4transkript - Website https://www.audiotranskription.de/f4

to allow the researcher to "focus on the interview content and the verbal prompts" [Ja14]. Based on the transcripts of the interviews, explanation building, i.e. an approach that tries "to "explain" a phenomenon . . . [by stipulating] a presumed set of causal links about it, or how or why something happened" [Yi09] was used to derive CSFs for Business-Managed IT systems. With the purpose of analyzing the data, the transcripts were coded by using the software f4analyse [7]. As a result, a more detailed description of the analyzed Business-Managed IT systems was derived and appropriate themes were created. These themes are presented in Section 5.

3.2 Critical Success Factors

The CSF method used in this paper was presumably first introduced by Rockart in 1979 and has evolved to be one of the most prevalent themes for the identification of factors for success in both research and practice. According to the definition by Rockart (1979) "critical success factors . . . are, for any business, the limited number of areas in which results, if they are satisfactory, will ensure successful competitive performance for the organization" [RA79]. In the context of this paper, a Business-Managed IT system was considered to be successful if the system was able to provide value for the business while at the same time complying with quality criteria for software systems. The criteria that was used to assess the quality of the studied software was derived from the ISO/IEC 25010 norm for Systems and Software Quality Requirements and Evaluation (2011). According to this norm, software product quality is achieved if the software possesses the following characteristics: *Functional Suitability, Performance Efficiency, Compatability, Usability, Reliability, Security, Maintainability* as well as *Portability* [IS10].
During the case study, it was left to the interview partners to determine whether the Business-Managed IT system was able to provide value to the business and did satisfy the quality criteria presented above. In case the aforementioned requirements were met, the system in question was considered to be a success.

4 Case Company and Business-Managed IT Systems

This paper conducted a case-study analysis of five Business-Managed IT systems at the KRONES corporation. KRONES is an Industrial Goods company which specializes on "lines for the beverage industry and food producers: process technology, filling technology, packaging machines . . . to IT solutions"[KR18]. The company currently employs ~14,500 employees [KR16] worldwide and generated a sales revenue of 3.4 billion € in the financial year 2016. Even though 90% of KRONES's revenue was generated outside of Germany, the majority of the production facilities with a workforce of over 10,000 people are still located in Germany.

[7] f4analyse - Website `https://www.audiotranskription.de/f4-analyse`

Name	Business Area	Location	Number of Users	Number of Developers	Critical for Business	Years of Existence
System A	Sales	Germany	~500	N/K	no	1 yr.
System B	Service	Belgium	124	1	yes	8 yrs.
System C	Manufacturing	Germany	~650	3	yes	7 yrs.
System D	Corporate Development	Worldwide	~300	N/K	no	1 yr.
System E	Manufacturing	Germany	N/K	1	no	6 mths.

Tab. 1: Key characteristics of studied Business-Managed IT systems at KRONES

Between October and December 2017, the author of the paper at hand performed a case study of five Business-Managed IT systems at KRONES and conducted interviews with stakeholders related to these systems. The systems subject to the analysis presented were already known to the IT department at KRONES since they were operated more or less "overtly" by the BUs and therefore account for Business-Managed IT. For every Business-Managed IT system, at least one key stakeholder from the responsible BU was interviewed. During the interviews, key characteristics for the studied systems were obtained from the interview partners. This data is shown in Table 1. All of the interview partners stated that they were very satisfied with the results accomplished by the respective Business-Managed IT systems. Based on the success characteristics proposed in Section 3.2, Table 2 classifies each system on whether it was able to provide value for the business and if an adequate level of Software Product Quality was met. [IS10] As the table shows, all of the analyzed Business-Managed IT systems at KRONES fulfil the proposed quality requirements for a successful Business-Managed IT system and even though some of the studied Business-

✓: Present ×: Not present	System A	System B	System C	System D	System E
Value for Business	✓	✓	✓	✓	✓
Functional Suitability	✓	✓	✓	✓	✓
Performance Efficiency	✓	✓	✓	✓	✓
Compatibility	✓	×	×	✓	✓
Usability	✓	✓	✓	✓	✓
Reliability	✓	✓	✓	✓	✓
Security	✓	✓	×	✓	✓
Maintainability	✓	✓	✓	✓	✓
Portability	✓	×	×	✓	✓

Tab. 2: Evaluation of Success of Business-Managed IT systems

Managed IT systems were not able to achieve satisfactory results regarding all of the quality characteristics described in Section 3.2 (such as *Compatibility* or *Portability*), the overall software quality of said systems can be considered to be sufficient for their use case.

5 Critical Success Factors Analysis

5.1 Research Results

This section describes the overall research results that were obtained from the case study conducted at KRONES. As a result from the analysis of the case study, nine CSFs for Business-Managed IT systems were obtained. They fall into three dimensions (*Project Dimension, Organizational Dimension* and *System Dimension*) and are presented in Table 3. The following sections contain parts of the original quotes from the interview partners and have been marked as such.

Project Dimension (P)	**Organizational Dimension (O)**	**System Dimension (S)**
P.1 Agile Development Methodology	O.1 Dedicated Product Team	S.1 Data Integration
P.2 High Management Commitment	O.2 Transparency	S.2 Compliance with Security Policies
P.3 User Participation and Involvement		S.3 Well Documented System Architecture
		S.4 Elaborate Vendor Management

Tab. 3: CSFs Overview

5.2 Project Dimension

The Project Dimension describes CSFs that relate to factors influencing the project methodology and characteristics of project management used for conducting the implementation of the Business-Managed IT system. Three CSFs are part of this category: (P.1) Agile Development Methodology, (P.2) Management Commitment, and (P.3) User Participation and Involvement.

P.1 Agile Development Methodology:
Two out of five studied projects deliberately chose an agile development methodology in which "parts of the software that was implemented were split up into smaller increments" (Systems A and C). The major advantage of this approach was that it "enabled the BUs to frequently develop prototypes which could be used by the key users to provide feedback to the development team" (System A). In addition to this, the software characteristics of *Functional Suitability* and *Usability* were also able to benefit heavily from using an agile development methodology. Moreover, the stated perception during the interviews was that following an agile methodology improved the overall flexibility of the BU, in particular because it allowed the developers to achieve a "high implementation speed" (Systems A, C and E) of the desired software.
P.2 High Management Commitment:
Newly implemented Business-Managed IT systems usually require some form of corporate investment, for example regarding either personnel- and / or financial resources. All of the studied five systems from the case study presented in Section 4 therefore sought to obtain the commitment of upper-level management representatives in a rather early stage of the system's implementation.
By obtaining management commitment, the interview partners stated that they were more likely to succeed "with allocation of new resources (e.g. project funding)" (System E) and in "convincing their end-users to use the newly developed software" (System B). In addition to this, support for the system amongst co-workers also increased because there was "more awareness about the project within the respective BU" (System A).
P.3 User Participation and Involvement:
To achieve success in Business-Managed IT systems, it also became apparent during the interviews that participation and involvement of users and other affected stakeholders of the system has to be practiced from early on by the people responsible for creating the Business-Managed IT systems.
Several interview partners stated that they were frequently meeting with their future users from the BUs to collect feedback and give their users updates about the current progress of the implementation. They also stated that "the barrier for interactions between the developers and the Business-Managed IT users is usually lower than in comparison to formal IT systems operated by the IT department" (System C).

5.3 Organizational Dimension

The Organizational Dimension describes CSFs which relate to the organizational circumstances under which the project is carried out. Two CSFs are part of this category: (O.1) Dedicated Product Team and (O.2) Transparency.

O.1 Dedicated Product Team:
As a major observation from this case study analysis, it was observed that the teams for Business-Managed IT systems at KRONES are usually assembled by key personnel from

the BUs themselves. All team members had a very specific background knowledge of the business processes in the BUs and, according to the interview partners, were thus able to make "carefully considered decisions regarding the utility of newly requested features" (System C).
Since the aforementioned projects were highly successful, the creation of dedicated, cross-functional product teams can be seen as a major factor that contributes to the overall success of Business-Managed IT systems. The formal objective of these product teams should not be to "deliver some piece of software which is then considered to be completed" [LF14], but instead to "own a product over its full lifetime" [LF14] where the capabilities of the team should ideally include "the full range of skills required for the development: user-experience, database, and project management" [LF14].

O.2 Transparency:
A common theme that was perceived in all of the analyzed case study interviews was the desire to be transparent towards both end-users as well as upper-level management about the activities of the Business-Managed IT systems. The reasoning of the interview partners in this particular case was that they perceived transparency to be useful because it allowed them to "collaborate more efficiently with other members of the team" (System A) and "to better understand the outcome of their actions as well as their influence on the overall performance of the project" (System C).
In particular, all of the five studied Business-Managed IT systems at KRONES tried to maintain a high level of transparency. One of the studied projects even made use of its own project management tool called *Jira* [8] (System A), a web-based project management software. Since every project member "can see what tasks are currently being worked on" (System A), the interview partner considered this to be a highly beneficial factor in creating a shared understanding of the desired system features amongst team members.

5.4 System Dimension

The System Dimension describes CSFs which relate to the technical characteristics of systems implemented by the Business-Managed IT systems. It consists of four CSFs: (S.1) Data Integration, (S.2) Security and Compliance, (S.3) System Architecture and (S.4) Vendor.

S.1 Data Integration:
A common observation from the Business-Managed IT systems that were analyzed at KRONES was that all of the implemented systems were accessing data stored in KRONES's Enterprise Resource Planning (ERP) system SAP. Data integration was of major importance for the implemented systems, and the design and quality of the system architecture heavily depended on having access to a dedicated interface for accessing and extracting data from the SAP system. In one case, the interface used to access KRONES's SAP system was described to be "rather shady" (System C). Most probably, the interview partner wanted

[8] Jira - Website `https://www.atlassian.com/software/jira`

to allude to the fact that a variety of other systems were involved in accessing the data in SAP, transforming it and finally storing it in the Business-Managed IT system. Due to these characteristics, data integration is considered a CSF for Business-Managed IT systems.

S.2 Compliance with Security Policies:

One of the product quality characteristics used for evaluating the success of Business-Managed IT systems was the criterion of *Security*. In the context of this study at hand, it is considered to be an important factor to take security and compliance into consideration when a new IT system is implemented decentrally in the BUs.

A positive example of an adequate assessment of security and compliance risks in a Business-Managed IT entity was the implementation of System A in which the responsible department did corporate closely with the IT department and external suppliers to ensure that all security and compliance requirements were met. Other projects however, especially those which were implemented more covertly and without the assistance from the IT department or external suppliers, failed to properly consider security and compliance risks and therefore pose a significant risk for the business which could result in data leakage and data loss.

As a result of these observations, Security and Compliance is considered to be a CSF for Business-Managed IT systems. Achieving adequate performance in this area makes the Business-Managed IT system highly beneficial for an organization and contributes positively towards the perceived success of the project, especially within the IT department.

S.3 Well Documented System Architecture:

Another CSF that was identified during the case study is the architecture of the newly implemented Business-Managed IT system. A well-designed and suitable system architecture usually results in a "low failure rate of the system" (System C) and increases the users' confidence towards the Business-Managed IT system. To achieve success of Business-Managed IT systems, it is therefore necessary to put sufficient effort into planning and elaborating the design of the system and profoundly documenting its architecture.

S.4 Elaborate Vendor Management:

During the interviews, the importance of an elaborate vendor management between KRONES and its suppliers was identified to be a CSF for Business-Managed IT systems. As stated by one interview partner, the fact that the "implemented solution was not specific to them" (System B) gave them the assurance that the software vendor had advanced knowledge regarding the implementation of the product and that KRONES could therefore benefit from lessons learned from the implementation of that product in other organizations.

6 Conclusion

6.1 Summary and Discussion

The case study presented in this paper at hand has theoretical implications for the research done in the field of Business-Managed IT and its impact on the ongoing "reorientation of IT" [Ko17] in corporations. To the best of knowledge of the author of this paper, no empirical research of CSFs in Business-Managed IT systems has been carried out so far. This present

paper therefore is the first publication to analyze this subject in an empirical study. As a result of the research at hand, it was shown that there are nine CSFs in Business-Managed IT systems which fall into three dimensions, namely *Project Dimension, Organizational Dimension* and *System Dimension*. In addition to this, the insights obtained from this research can be used by both academia and practitioners to evaluate the "controlled use of Business-Managed IT" [KWS17] in organizations.
The CSFs presented in this research were obtained from an empirical study of Business-Managed IT activities and it is possible that substitutes for the CSFs exist. Regarding for example the CSF "P.2 - High Management Commitment", it is questionable whether support by the management really fosters the acceptance of the system by the employees or if the same level of support could also be obtained by using other methods to raise awareness.

6.2 Research Limitations

Several limitations apply to the research presented in this paper. First, the case study that was used to derive CSFs in Business-Managed IT systems was exclusively focused on only one corporation. KRONES is an Industrial Goods company from Germany which operates its business from its corporate headquarters in Neutraubling. Even though the case study that was conducted during this research also analyzed systems that were located in other subsidiaries of KRONES (e.g. in Belgium), it can not be denied that the company is operating with a certain influence from German culture and mentality. This could have influenced the perception of successful leadership in Business-Managed IT systems and might also have biased the interview partners to stress certain aspects of the Project Dimension and Organizational Dimension of the CSFs for Business-Managed IT systems more than others. As part of the case study, only systems which were already known to the IT department at KRONES were studied. These systems, by their nature, are relatively *overt* in the organization and might therefore not allow to draw conclusions about the nature of *covert* Business-Managed IT systems which are operated without any knowledge from external departments.

6.3 Future Work

The main objective of this paper was to study CSFs in Business-Managed IT. For future research, more work is required with regard to the procedures of identifying Business-Managed IT applications in organizations. A variety of publications were presented that evaluate and assess the criticality of the identified Business-Managed IT applications, but unfortunately only relatively little information is available regarding the first initial identification of applications and systems operated covertly within the BUs.
One approach was proposed by Fürstenau and Rothe whom employed network analysis visualizations to develop "a method to identify Shadow-IT system's [sic] and assess their importance with respect to their architectural embeddedness" [FR14]. The authors

followed a manual procedure which required skilled professionals with "business analysis competencies [...] to align IT and business perspective" [FR14]. Future work in this area could investigate the feasibility of automatically identifying Business-Managed IT systems within the network. Such an approach could leverage existing network scanning tools such as *nmap*[9] in combination with network traffic analysis techniques to automatically discover new systems which are deployed covertly in the BUs.

Acknowledgement

The author would like to thank his supervisor, Prof. Dr. Markus Westner, for his comprehensive feedback and encouragement. Special thanks also go to KRONES and the interview partners for enabling this research.
The author would furthermore like to thank the anonymous reviewers for their feedback.

References

[BB94] Byers, C. R.; Blume, Debbie: Tying critical success factors to systems development. Information & Management, 26(1):51–61, 1994.

[CC08] Chow, Tsun; Cao, Dac-Buu: A survey study of critical success factors in agile software projects. Journal of Systems and Software, 81(6):961–971, 2008.

[CSC14] Chua, Cecil; Storey, Veda; Chen, Langtao: Central IT or Shadow IT? Factors shaping users' decision to go rogue With IT. In (Myers, Michael D.; Straub, Detmar W., eds): Proceedings of the International Conference on Information Systems - Building a Better World through Information Systems, International Conference on Information Systems (ICIS) 2014, Auckland, New Zealand, December 14-17, 2014. Association for Information Systems, 2014.

[EG07] Eisenhardt, Kathleen M.; Graebner, Melissa E.: Theory building from cases: Opportunities and challenges. Academy of Management Journal, 50(1):25–32, 2007.

[FR14] Fürstenau, Daniel; Rothe, Hannes: Shadow IT systems: Discerning the good and the evil. In: 22st European Conference on Information Systems, ECIS 2014, Tel Aviv, Israel, June 9-11, 2014. European Conference on Information Systems, 2014.

[Hu16] Huber, Melanie; Zimmermann, Stephan; Rentrop, Christopher; Felden, Carsten: The Relation of shadow systems and ERP systems—Insights from a multiple-case study. Systems, 4(1):11, 2016.

[IS10] ISO - International Organization for Standardization: , ISO/IEC 25010 - Systems and software engineering - Systems and software Quality Requirements and Evaluation (SQuaRE) - System and software quality models, 2010.

[Ja14] Jamshed, Shazia: Qualitative research method-interviewing and observation. Journal of basic and clinical pharmacy, 5(4):87–88, 2014.

[9] nmap Project - Website `https://nmap.org/`

[Ko17] Kopper, Andreas: Perceptions of IT managers on Shadow IT. In: 23rd Americas Conference on Information Systems, AMCIS 2017, Boston, MA, USA, August 10-12, 2017. Association for Information Systems, 2017.

[Ko18] Kopper, Andreas; Fürstenau, Daniel; Zimmermann, Stephan; Rentrop, Christopher; Rothe, Hannes; Strahringer, Susanne; Westner, Markus: Business-Managed IT: A conceptual framework and empirical illustration. In: 26th European Conference on Information (ECIS) 2018, Portsmouth, UK, June 23-28, 2018. European Conference on Information Systems, 2018.

[KR16] KRONES AG - Annual Report 2016, https://www.krones.com/media/downloads/GB_2016_AG_e.pdf. (Accessed on 07/04/2018).

[KR18] KRONES - we do more, https://www.krones.com/en/index.php. (Accessed on 07/04/2018).

[KW16a] Kopper, Andreas; Westner, Markus: Deriving a framework for causes, consequences, and governance of Shadow IT from literature. Multikonferenz Wirtschaftsinformatik - MKWI, 3:1687–1699, 2016.

[KW16b] Kopper, Andreas; Westner, Markus: Towards a taxonomy for Shadow IT. In: 22nd Americas Conference on Information Systems, AMCIS 2016, San Diego, CA, USA, August 11-14, 2016. Association for Information Systems, 2016.

[KWS17] Kopper, Andreas; Westner, Markus; Strahringer, Susanne: Kontrollierte Nutzung von Schatten-IT. HMD Praxis der Wirtschaftsinformatik, 54(1):97–110, 2017.

[LE06] Levy, Yair; Ellis, Timothy J.: A systems approach to conduct an effective literature review in support of Information Systems research. Informing Science, 9:181–212, 2006.

[LF14] Microservices: A definition of this new architectural term, https://martinfowler.com/articles/microservices.html. (Accessed on 07/04/2018).

[RA79] Rockart, J. F.; Administration, Harvard University Graduate School of Business: Chief executives define their own data needs. Harvard Business School, 1979.

[SB14] Silic, Mario; Back, Andrea: Shadow IT – A view from behind the curtain. Computers & Security, 45:274–283, 2014.

[UA16] Urbach, Nils; Ahlemann, Frederik: IT-Management im Zeitalter der Digitalisierung. Springer, Berlin, 2016.

[Yi09] Yin, R. K.: Case Study Research: Design and Methods. Applied Social Research Methods. SAGE Publications, Thousand Oaks, 2009.

[ZRF14] Zimmermann, Stephan; Rentrop, Christopher; Felden, Carsten: Managing Shadow IT instances - A method to control autonomous IT solutions in the business departments. In: 20th Americas Conference on Information Systems, AMCIS 2014, Savannah, Georgia, USA, August 7-9, 2014. Association for Information Systems, 2014.

[ZRF16] Zimmermann, Stephan; Rentrop, Christopher; Felden, Carsten: Governing identified shadow IT by allocating IT task responsibilities. In: 22nd Americas Conference on Information Systems, AMCIS 2016, San Diego, CA, USA, August 11-14, 2016. Association for Information Systems, 2016.

[ZRF17] Zimmermann, Stephan; Rentrop, Christopher; Felden, Carsten: A multiple case study on the nature and management of shadow Information Technology. Journal of Information Systems, 31(1):79–101, 2017.

Der Einfluss der strategischen Rolle der IT auf die IT-Strategieentwicklung

Jenny Schwarz[1]

Abstract: Welche Rolle die IT in einem Unternehmen einnimmt, wird heute in der Praxis unter den Begriffen ‚bimodale IT' oder ‚Two-Speed-IT' diskutiert. Darunter wird verstanden, dass die IT sowohl eine traditionelle als auch eine schnellere Rolle in einem Unternehmen annehmen kann. Abhängig von der strategischen Rolle besitzt die IT unterschiedliche Eigenschaften und benötigt eine dementsprechende IT-Strategie. Daneben gibt es ein weiteres Modell, das ‚Strategic Grid', welches die IT sogar in vier unterschiedliche Rollen unterscheidet. Ziel dieser Arbeit ist es, mithilfe von sechs IT-Handlungsfeldern zu zeigen, ob eine Unterscheidung der IT in zwei Rollen ausreichend ist, oder ob die detailliertere Differenzierung der Rolle der IT nach dem Strategic Grid für die IT-Strategieentwicklung notwendig ist. So kann geklärt werden auf welcher Grundlage Unternehmen ihre IT-Strategie aufbauen sollten.

Keywords: Strategische Rolle der IT, Strategic Grid, IT-Management, IT-Strategie, strategische Handlungsfelder

1 Einleitung

Praktisch jedes Unternehmen ist digital, denn die IT ist allgegenwärtig und in unterschiedlicher Intensität in das tägliche Unternehmensgeschäft integriert [KNP17]. Die Intensität des IT-Einsatzes ist teilweise bereits so hoch, dass Unternehmen abhängig von einem zuverlässigen IT-Betrieb und ohne IT nicht mehr überlebensfähig sind. Folglich wird durch die IT die Wettbewerbsfähigkeit eines Unternehmens sichergestellt. [Fo16] Daneben haben viele Unternehmen erkannt, dass IT-Innovationen einen Wettbewerbsvorteil ausmachen können, und setzten zunehmend Digitalisierungsinitiativen, welche mehr IT in den Unternehmensalltag bringen, um. Dabei sind viele Unternehmen jedoch weder strukturell noch prozessual auf den schnellen Wandel der IT vorbereitet. Ist ein Unternehmen nicht dazu in der Lage in Zusammenarbeit von Geschäft und IT Innovationen zu realisieren und umzusetzen, stellt dies ein großes Hindernis für die Digitale Transformation dar, womit der Wandel von Geschäftsmodellen durch die IT bezeichnet wird. [UA17] Grund hierfür ist häufig das Fehlen eines unternehmensweiten IT-Managements, welches essenziell für die Erhaltung der Wettbewerbsfähigkeit und die Realisierung von Wettbewerbsvorteilen ist [MB15]. Ein integraler Bestandteil und Erfolgsfaktor des IT-Managements ist die Definition einer passenden IT-Strategie. Wichtig dabei sind die Identifikation der strategischen Rolle der IT und das Verständnis, wie diese Rolle die IT-Strategie beeinflusst. [PS13] Heute steht die IT in vielen Unterneh-

[1] HTWG Konstanz, Fakultät Informatik, Alfred-Wachtel-Straße 8, 78462 Konstanz, jenny.schwarz@bitco3.com

men vor der Herausforderung einen verlässlichen IT-Betrieb zur Sicherung der Wettbewerbsfähigkeit zu gewährleisten und gleichzeitig die Entwicklung von Innovationen zu unterstützen, sodass Wettbewerbsvorteile umgesetzt werden. Dies wird als ‚bimodale IT' oder ‚Two-Speed-IT' bezeichnet, wobei eine Differenzierung zwischen einer stabilen klassischen und einer schnellen IT erfolgt. [Ti17] Neben der Abgrenzung der IT in zwei Rollen existiert ein Model, welches als ‚Strategic Grid' bezeichnet wird und die strategische Rolle der IT durch vier verschiedene Modi beschreibt. Diese werden als Support-, Fabrik-, Umstrukturierungs- und strategischer Modus bezeichnet und unterscheiden sich hinsichtlich ihrer Eigenschaften, welche durch die Kombination aus aktuellem und zukünftigem Einfluss der IT auf das Geschäft resultieren. Infolgedessen haben die Modi unterschiedliche Anforderungen an das IT-Management. [NM05] Daraus ergibt sich nun die Frage, ob die Unterscheidung in zwei strategische Rollen der IT für die IT-Strategieentwicklung ausreichend ist.

In der Literatur bestehen diese Sichtweisen nebeneinander, ohne dass der Widerspruch zwischen diesen ausreichend geklärt wurde. Ziel dieser Forschungsarbeit ist es diese Lücke zu schließen. Dabei soll herausgearbeitet werden, ob es sinnvoll ist IT als Two-Speed-IT zu definieren, oder ob eine stärkere Differenzierung in vier verschiedene Rollen notwendig ist.

Hierfür werden zunächst die Grundlagen zur strategischen Rolle der IT und den strategischen Handlungsfeldern des IT-Managements in Kapitel 2 erläutert. In Kapitel 3 ist die Forschungsmethodik des Literaturreview beschrieben. Anschließend werden die Forschungsergebnisse in Kapitel 4 vorgestellt und in Kapitel 5 diskutiert. Der Artikel schließt mit einem Fazit in Kapitel 6.

2 Grundlagen

Zur Erreichung dieses Forschungsziels werden zwei Aspekte betrachtet. Zum einen geht es hierbei um die Frage nach der strategischen Rolle der IT und zum anderen, um die Handlungsfelder im Bereich IT-Management, die sich an der strategischen Rolle der IT ausrichten.

2.1 Die strategische Rolle der IT

Unternehmen stehen heute vor der Herausforderung die Wettbewerbsfähigkeit durch eine traditionelle IT sicherzustellen und gleichzeitig mit der agilen IT Wettbewerbsvorteile zu identifizieren und umzusetzen. Damit existiert eine Dualität der IT und es muss ein Spagat zwischen der traditionellen und der agilen IT ausgeführt werden. Dies wird unter dem Begriff ‚Two-Speed-IT' erörtert. Dabei gilt es in der traditionellen IT den zuverlässigen IT-Betrieb und die Ausfallsicherheit von Systemen sicherzustellen. Die agile IT hingegen wird durch die Umsetzung von Digitalisierungsmaßnahmen und den Betrieb von Innovationen beherrscht. [Ti17]

Neben dieser Betrachtungsweise auf die strategische Rolle der IT existiert ein weiteres Modell, welches die Rolle der IT weiter ausdifferenziert. Das Strategic Grid von Nolan und McFarlan definiert die strategische Rolle der IT durch die Betrachtung von zwei Aspekten. Zunächst werden der zukünftige Einfluss und die Chancen der IT bewertet. Dadurch lässt sich die IT in eine offensive und eine defensive IT unterscheiden. Dabei gilt es für die offensive IT aggressiv im Wettbewerb aufzutreten, um Chancen zu nutzen und Wettbewerbsvorteile auszubauen. Die defensive IT hingegen liefert einen geringen zukünftigen Einfluss auf das Unternehmen und sollte daher eher Kosteneffizienz verfolgen und ununterbrochene Verfügbarkeit der IT-Systeme liefern. [NM05] Soweit lässt sich dies mit der Two-Speed-IT vergleichen. Zusätzlich wird hier allerdings der aktuelle Einfluss der IT auf das Unternehmen betrachtet, wobei es sich um die Notwendigkeit der Ausfallsicherheit und damit der Minimierung von Risiken handelt. Damit wird deutlich, dass in einem Unternehmen sowohl IT-Systeme mit geringer aktueller Relevanz für das Unternehmen existieren, als auch IT-Systeme deren Ausfall einen beträchtlichen unternehmerischen Schaden verursachen können, da das Geschäft kritisch von diesen Systemen abhängig ist. [Do09] Basierend auf diesen beiden Aspekten, lassen sich vier strategische Rollen mit unterschiedlichen Eigenschaften ableiten, wie in Abb. 1 dargestellt.

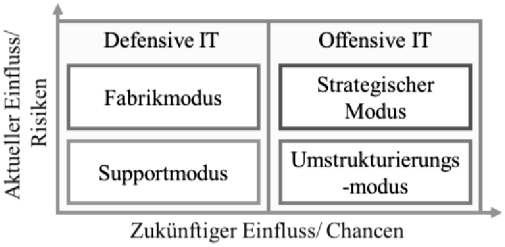


Abb. 1: Strategic Grid in Anlehnung an [Do09]

- Supportmodus: IT-Systeme welche dem Supportmodus zugeordnet werden, haben einen geringen aktuellen Einfluss sowie einen geringen zukünftigen Einfluss auf das Geschäft. Aus diesem Grund agieren sie defensiv als Technologie-Follower. Da hier vor allem Standardsysteme implementiert werden, wird dieser Bereich der IT auch als „Commodity-IT“ bezeichnet. [Do09]
- Fabrikmodus: Der zukünftige Einfluss der IT-Systeme im Fabrikmodus ist zwar gering für das Geschäft, doch der aktuelle Einfluss dagegen ist sehr hoch. Deshalb wird hier die Einführung von stabilen und zuverlässigen IT-Systemen gefordert [Do09], deren Ausfallsicherheit gewährleistet werden kann. Dies ist besonders wichtig, da Unternehmen kritisch von den hier angesiedelten IT-Systemen abhängig sind und ein Ausfall extreme Folgen mit sich bringen kann [NM05].
- Strategischer Modus: Hier sind sowohl der zukünftige als auch der aktuelle Einfluss der IT auf das Geschäft sehr hoch. Folglich gilt hier, ähnlich im Fabrikmodus, dass das Geschäft kritisch von der IT abhängig ist und deshalb ein ununterbrochener IT-

Betrieb sichergestellt werden muss. [NM05] Gleichzeitig gilt es aber auch darum Wettbewerbsvorteile zu verfolgen und Innovationen umzusetzen [Do09].

- Umstrukturierungsmodus: Die IT-Systeme im Umstrukturierungsmodus haben zwar einen hohen zukünftigen Einfluss, doch nur einen geringen aktuellen Einfluss. Da das Geschäft hier nicht kritisch von der IT abhängig ist, stehen hier der Innovationsbedarf und die Realisierung von Wettbewerbsvorteilen im Fokus. [NM05]

2.2 Handlungsfelder der IT-Strategie

Die Hauptaufgabe der IT-Strategie ist die Gestaltung der IT in Abstimmung auf die Unternehmensstrategie. Ein Unternehmen muss dazu in der Lage sein, die richtigen Entscheidungen in Bezug auf IT-Themen zu treffen und dabei die Unternehmenssituation miteinzubeziehen. Da die IT mittlerweile zu einem wichtigen Indikator für den Unternehmenserfolg geworden ist, ist auch das IT-Management und die Entwicklung einer IT-Strategie ein kritischer Erfolgsfaktor. [FG07] Dabei muss die Gestaltung der IT hinsichtlich der folgenden Themengebiete geklärt werden:

- IT-Governance: Durch die IT-Governance soll die Zusammenarbeit und das Alignment zwischen der IT und dem Geschäft gefördert werden. Hierfür werden Strukturen, Prozesse und relationale Mechanismen definiert. [HG15] Dabei gilt es festzustellen inwieweit eine zentrale oder dezentrale IT-Governance benötigt wird. Dies ist abhängig von der Zielsetzung, denn eine zentrale IT-Governance unterstützt bei der Umsetzung von Standards während durch eine dezentrale IT-Governance die Bedürfnisse des Geschäfts besser befriedigt werden können [PS13].

- IT-Architektur: Mithilfe einer IT-Architektur kann eine Gesamtsicht auf das Unternehmen geschaffen werden, welche alle wesentlichen Geschäfts und IT-Strukturen und deren Verknüpfungen aufzeigt. So werden Zusammenhänge deutlich und Abhängigkeit und Auswirkungen von Veränderungen werden transparent. Auf Grund der hohen Anzahl an verschiedenen Technologien und Entwicklungen in einem Unternehmen muss die Komplexität der IT-Architektur eingedämmt werden. Dies kann durch die Einführung von Standards umgesetzt werden, wodurch gleichzeitig die Einführung neuer Technologien und die Umsetzung von Skaleneffekten erleichtert wird. [Ha09] Folglich beschäftigt sich die IT-Architektur mit dem Grad der Standardisierung.

- IT-Sourcing: Unter IT-Sourcing wird die Beschaffung von IT-Produkten und IT-Dienstleistungen verstanden, welche sowohl durch interne Leistungserstellung als auch von externen Dienstleistern bezogen werden kann [GM14]. Hierbei stellt sich die Frage nach dem Grad des IT-Outsourcings, also wieviel der IT nach außen vergeben wird und wie viel selbst betrieben wird.

- IT-Organisation: Aufgabe der IT-Organisation ist es die IT im Unternehmen einzubetten und Entscheidungsrechte und Verantwortlichkeiten richtig zu positionieren [PS13]. Dabei muss festgestellt werden, ob es sinnvoll ist eine zentrale IT-Abteilung

zu haben, welche unter Vorgaben arbeitet, oder ob dezentrale IT-Einheiten notwendig sind welchen mehr Freiräume geboten werden [Ha09]. Es gilt also zu klären, ob die IT als eigenständiger Bereich, als Stabstelle oder unter einer Matrixorganisation geführt wird und wie groß die Freiräume sind unter welchen die IT arbeitet [Al16].

- IT-Personal: Das Wissen des IT-Personals ist eine zentrale Ressource, denn die individuellen Kenntnisse und Fähigkeiten des IT-Personals führen zu überlegenen Geschäftsprozessen und damit schlussendlich zu Wettbewerbsvorteilen [St08]. Gleichzeitig veraltet Wissen heute sehr schnell, da ständig neue technologische Fortschritte auftreten. Folglich wird IT-Personal mit den passenden Kompetenzen gesucht und muss permanent weiterqualifiziert werden. [Jo14]
- IT-Sicherheit: Die IT-Sicherheit muss das Unternehmen vor unterschiedlichen IT-Sicherheitsrisiken schützen [HS10]. Allerdings kann die Einführung von IT-Sicherheitsmaßnahmen teuer werden und zusätzlich die Nutzung eines Systems maßgeblich einschränken. Herauszufinden welche Sicherheitsanforderungen tatsächlich bestehen, ist folglich die zentrale Herausforderung der IT-Sicherheit. [GM17]

3 Methodik

Als Grundlage zur Erreichung des Forschungsziels dient die Methode des Literaturreviews. Die hierfür gesammelte Literatur, welche inhaltlich die sechs strategischen Handlungsfelder behandelt, bezieht sich auf Veröffentlichungen bis Ende August 2017. Diese stammt aus der Hochschulbibliothek der HTWG Konstanz, der Onlineplattform IEEE und Google Scholar. Dabei behandelt die gesammelte Literatur Bücher und Artikel welche im Allgemeinen die Themen IT-Management und IT-Strategie thematisieren. Diese sind gleichgewichtet und gilt es im nächsten Schritt auf Aussagen zu untersuchen, welche beschreiben was von den strategischen Handlungsfeldern hinsichtlich der Eigenschaften der Modi erwartet wird. Als Hilfestellung werden die Eigenschaften der vier verschiedenen strategischen Rollen der IT aus dem Strategic Grid als Parameter definiert, um als Schlüsselwörter für das Literaturreview zu fungieren [SLT09]. Für den Supportmodus, welchem Standradsysteme und Commodity-IT zugeordnet werden, werden die Schlüsselwörter *Standards, Redundanzen, Synergien, Kosten, teuer* und *Skaleneffekte* definiert. Durch den geringen Innovationsgrad im Support- und Fabrikmodus, sind diese beiden Modi *statisch*, *defensiv* und durch *Routine* geprägt. Diese Eigenschaften werden als Schlüsselwörter für den Supportmodus und gleichzeitig für den Fabrikmodus identifiziert. Auf Grund der hohen Risiken im Fabrikmodus und im strategischen Modus können die Schlüsselwörter *Kontrolle, Sicherheit*, *Risiken, Bedürfnisse, Verlässlichkeit* und *sensible* oder *kritische Geschäftsprozesse* abgeleitet werden. Daneben teilt der strategische Modus Schlüsselwörter mit dem Umstrukturierungsmodus, da in beiden Modi hohe Innovationspotenziale und Chancen bestehen. Die Schlüsselwörter sind daher *Innovation, Differenzierung, Dynamik*, *Digitalisierung, Anpassungsfähigkeit, Agilität*

und *schnelle Veränderung*. Anhand dieser Parameter wird die Literatur hinsichtlich der sechs strategischen Handlungsfelder im Bereich IT-Management untersucht.

Zur Analyse der gesammelten Informationen wird die Forschungsmethode ‚Coding' umgesetzt [SLT09]. Dazu werden im ersten Schritt die Daten aus dem Literaturreview zu den einzelnen strategischen Handlungsfeldern, welche über die definierten Parameter erhoben werden, analysiert. Jedes Datum trifft eine Aussage darüber, welche Anforderungen basierend auf den definierten Parameter hinsichtlich eines Handlungsfeldes auftreten. Diese Aussagen werden dann einem oder mehreren Modi zugeordnet, abhängig davon wessen Eigenschaften durch den Parameter beschrieben werden. So entsteht eine Sammlung von Informationen über jede der vier strategischen Rollen. Anhand dieser Sammlung von Informationen kann festgestellt werden, welches tendenzielle Verhalten von den strategischen Handlungsfeldern hinsichtlich der strategischen Rollen erwartet wird. Im zweiten Schritt werden die Ergebnisse nach den Modi sortiert, sodass sichtbar wird, welche Anforderungen eine strategische Rolle der IT entsprechend ihrer Eigenschaften an die Handlungsfelder stellt. Als letztes soll versucht werden die vier Modi zu zwei Modi zusammenzufassen, indem die Anforderungen miteinander verglichen werden. An dieser Stelle soll sich zeigen, ob das Model der Two-Speed-IT auseichend ist, oder, dass das Strategic Grid eine bessere Beschreibung der strategischen Rolle der IT liefert.

4 Ergebnisse

Basierend auf dem Literaturreview haben sich folgende Sachverhalte ergeben, welche in den Netzdiagrammen in Abb. 2 dargestellt sind. Die Netzdiagramme zeigen welche Anforderungen an die verschiedenen strategischen Handlungsfelder hinsichtlich der vier Modi in der Literatur bestehen. Dabei wurde die Wertigkeit über die Quantität der Aussagen aus der Literatur abgeleitet. Ausgehend von den sechs strategischen Handlungsfeldern wird im Folgenden anhand von Textbeispielen aus der Literatur besprochen, welche Ergebnisse sich für die vier Modi ergeben haben.

Die Aussage „[…] dass IT-Innovationen idealerweise dort entstehen sollten, wo sie später auch zum Einsatz kommen werden – nämlich in den Fachabteilungen." [UA16] zeigt, dass sowohl im strategischen als auch im Umstrukturierungsmodus vor allem dezentrale IT-Governance-Strukturen benötigt werden. Allerdings können zentrale Strukturen mehr Kontrolle und Steuerbarkeit bieten, was im strategischen und im Fabrikmodus sicherzustellen ist [PS13]. Da das Geschäft im Support- und Fabrikmodus eher statisch ist und nicht durch permanente Veränderungen geprägt ist, eignet sich hier die Nutzung von Standards und die Zentralisierung von IT-Leistungen [FG07]. Dennoch müssen auch dezentrale Strukturen für den Fabrikmodus zugelassen werden, um den Bedürfnissen des Geschäfts entgegenkommen zu können [PS13].

„Standardisierung ermöglicht es einem Unternehmen Kosten einzusparen und wiederholbare Ergebnisse zu erzielen. Eine nicht so stark integrierte IT-Architektur ist dagegen

flexibler und kann schneller und risikoloser verändert werden.“ [Re13] Folglich wird im Supportmodus eine hohe Standardisierung der IT-Architektur verfolgt. Währenddessen gilt es in den anderen Modi auch darum einen gewissen Grad an Heterogenität zuzulassen, um mehr Flexibilität und Reaktionsfähigkeit gegenüber sich ändernde Anforderungen zu erlangen.

Mithilfe von Aussagen aus der Literatur, wie beispielsweise “[…] outsourcing as a way of cutting costs.“ [WP02], konnte entschieden werden, dass im Supportmodus ein sehr hohes Potenzial für Outsourcing besteht. Wenn jedoch die betroffenen Prozesse von hoher strategischer Bedeutung für ein Unternehmen sind, sollte eher auf Insourcing zurückgegriffen werden [GM14]. Folglich sollten sowohl im Fabrik—als auch im strategischen Modus die Frage nach dem IT-Outsourcing gut überdacht werden. Durch das Ziel der „[…] Sicherstellung auch zukünftiger Wettbewerbsvorteile durch die Definition so genannter Kernkompetenzen und Kernleistungen, die aktuell und zukünftig von so hoher Bedeutung sind, dass sie nicht fremd vergeben werden dürfen“ [KWW09] wird klar, dass das IT-Outsourcing auch im Umstrukturierungsmodus nicht zu voreilig umgesetzt werden sollte. Gleichzeitig kann IT-Outsourcing aber auch Vorteile bieten, da dadurch Know-How und Kompetenzen des Lieferanten so zugänglich werden [Jo14].

Der Themenbereich IT-Organisation verhält sich im Netzdiagramm identisch zur IT-Governance. Denn „Feste Strukturen in der IT erlauben effiziente Arbeitsabläufe und fördern die Automatisierung, stoßen aber bei einer Forcierung der Innovationstätigkeit an ihre Grenzen.“ [UA16] Folglich unterstützen strenge Vorgaben den Supportmodus dabei Kosten einzusparen und Skaleneffekte zu realisieren. Für den strategischen und den Umstrukturierungsmodus hingegen ist es wichtig Freiräume zuzulassen, um die Entwicklung von Innovationen und die Kreativität der Mitarbeiter zu fördern. „Autonomie […] führt zu einer passgenaueren Unterstützung und einer größeren Flexibilität bei der Umsetzung der Geschäftsanforderungen für die Geschäftseinheit.“ [Ha14] Um flexibel auf Veränderungen im Fabrikmodus reagieren zu können und die Fachbereiche bestmöglich unterstützen zu können, sind Freiräume essenziell. Doch beutetet dies gleichzeitig das Kontrolle und Steuerbarkeit abgegeben werden. Aus diesem Grund ist es notwendig auch feste Vorgaben im strategischen und Fabrikmodus umzusetzen.

Die Aussage „Es sind die individuellen Fertigkeiten und Kenntnisse der Mitarbeiter, die zu überlegenen Geschäftsprozessen führen und damit Wettbewerbsvorteile begründen“ [St08] verdeutlicht, dass sowohl im strategischen als auch im Umstrukturierungsmodus von den Mitarbeitern hohe IT-Kompetenzen abverlangt werden. Zusätzlich kommt beim strategischen Modus hinzu, dass die Ausfallsicherheit von Systemen und ein ununterbrochener zuverlässiger IT-Betrieb sichergestellt werden müssen. Folglich sind auch im Fabrikmodus die Anforderungen an die Mitarbeiter hoch. Im Supportmodus werden stattdessen zunehmend IT-Managementkompetenzen benötigt, da durch das hohe IT-Outsourcing-Potenzial besonders Lieferantenbeziehungen gepflegt werden müssen [Jo14].

An das letzte Handlungsfeld IT-Sicherheit bestehen im Allgemeinen sehr hohe Anforderungen. Die meisten IT-Systeme arbeiten nicht losgelöst voneinander, oder sind über Netzwerke miteinander verbunden, was das Ziehen einer Grenze zwischen den IT-Systemen und deren Sicherheitsanforderungen erschwert. Trotzdem haben sich kleine Unterschiede gezeigt. „Denn Sicherheit ist auf der einen Seite sehr kostspielig, auf der anderen Seite kann sie die Nutzung eines Services erheblich einschränken.“ [GM17] Da der Supportmodus einen Fokus auf Kosten setzt, fallen die IT-Sicherheitsanforderungen geringer aus als im strategischen und Fabrikmodus. Hier kann eine IT-Sicherheitslücke fatale Folgen für ein Unternehmen bewirken. Des Weiteren können IT-Sicherheitsmaßnahmen Einschränkungen verursachen, was die Realisierung von Innovationen beeinträchtigen kann. Die Anforderungen im Umstrukturierungsmodus fallen daher ebenfalls zurück, wobei im strategischen Modus die Kritikalität der IT-Systeme so hoch ist, dass die Erfüllung von IT-Sicherheitsanforderungen überwiegt.

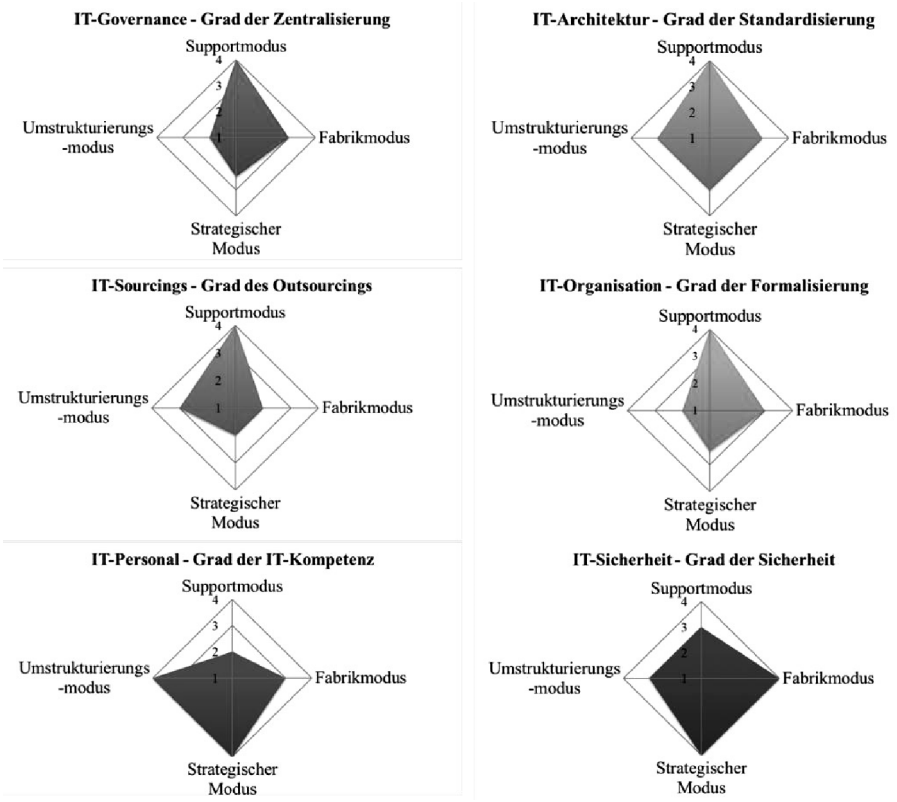


Abb. 2: Anforderungen an die strategischen Handlungsfelder hinsichtlich des Strategic Grids

5 Diskussion

Basierend auf den vorliegenden Ergebnissen ist nun zu fragen, ob es sinnvoll ist die vier Modi zusammenzufassen, sodass von zwei strategischen Rollen der IT gesprochen werden kann. Wie unter den Grundlagen bereits erläutert wurde, werden der Support- und der Fabrikmodus als defensive IT bezeichnet und der strategische und der Umstrukturierungsmodus als offensive IT. Lassen sich die vier Modi dementsprechend zusammenfassen, liegen nur noch zwei strategische Rollen der IT vor, welche mit den beiden Rollen aus der Two-Speed-IT übereinstimmen. Um nun feststellen zu können, ob sich die Modi zusammenfassen lassen, gilt es als nächstes die ermittelten Anforderungen an die sechs verschiedenen Handlungsfelder aus Abbildung 2 nach den vier Modi zu sortieren und auf Gemeinsamkeiten zu untersuchen. In Abbildung 3 sind die Anforderungen der Modi aus der defensiven IT übereinandergelegt. Dabei ist zu erkennen, dass große Diskrepanzen zwischen den Anforderungen der beiden Modi bestehen.

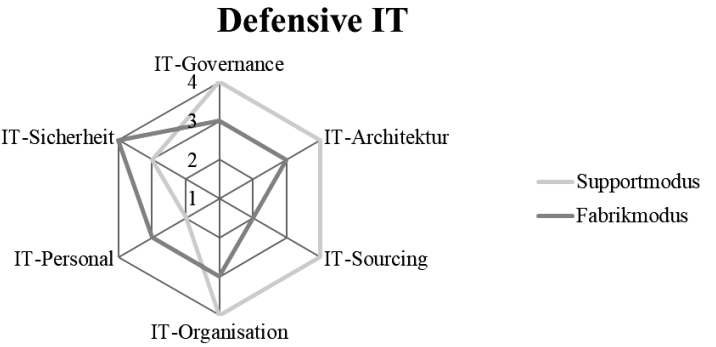


Abb. 3: Vergleich Support- und Fabrikmodus

Es fällt auf, dass die Netzdiagramme des Support- und des Fabrikmodus sehr unterschiedlich aussehen und sich in nicht einem Punkt treffen. Obwohl beide Modi keinen zukünftigen Einfluss auf ein Unternehmen haben und defensiv im Wettbewerb auftreten, sind die Anforderungen doch sehr verschieden. Grund hierfür ist der hohe aktuelle Einfluss, welchen der Fabrikmodus auf das Geschäft ausübt. Da hier ein Ausfall eine Existenzbedrohung bedeuten kann, werden trotz hoher Kosten notwendige IT-Sicherheitsmaßnahmen umgesetzt. Des Weiteren können dezentrale IT-Governance-Strukturen, ein gewissen Maß an heterogener IT-Architektur und Freiräumen in der IT-Organisation dabei unterstützen die Bedürfnisse des Geschäfts bestmöglich zu erfüllen. Hierfür wird entsprechendes IT-Personal benötigt, welches die Kompetenz besitzt auf die Bedürfnisse des Geschäfts zu reagieren und eine ununterbrochene Verfügbarkeit der IT sicherstellen kann. Im Supportmodus hingegen wird darauf geachtet, Kosten durch die Umsetzung von Standards und Skaleneffekten einzusparen. Dieser Unterschied wirkt sich logischerweise auf die Anforderungen aus. Zentrale IT-Governance, standardisierte IT-Architektur und formalisierte IT-Organisation helfen dabei, Standards und Skalenef-

fekte zu realisieren. Dies wiederum begünstigt die Umsetzung von IT-Outsourcing-Vorhaben. Folglich werden auch andere IT-Personalkompetenzen gefordert. Im Fabrikmodus wird eine hohe IT-Sicherheit zur Risikominimierung gefordert während die Systeme im Supportmodus weniger kritisch für das Unternehmen sind.

Auch die Netzdiaramme der offensiven IT aus Abb. 4 zeigen sich sehr verschieden, wenngleich diese sich immerhin in zwei Punkten treffen. Sowohl der strategische als auch der Umstrukturierungsmodus setzten auf sehr hohe IT-Personalkompetenzen, um neue Innovationen entwickeln und realisieren zu können. Auch im Bereich IT-Architektur bestehen dieselben Anforderungen, da Standardisierung die Digitalisierung und die Einführung neuer Innovationen erleichtern und unterstützen kann, gleichzeitig wird aber auch Heterogenität für die Entwicklung neuer Innovationen benötigt. Bezüglich der weiteren Handlungsfelder bestehen allerdings Diskrepanzen hinsichtlich der Anforderungen. Auch hier ist es auf den aktuellen Einfluss, welchen der strategische Modus auf das Geschäft hat, zurückzuführen. Folglich werden im strategischen Modus mehr zentrale IT-Governance-Strukturen und Vorgaben in der IT-Organisation benötigt, um die Steuerbarkeit und Kontrollierbarkeit der IT zu erhöhen. Der Verlust von Steuerbarkeit und Kontrollierbarkeit tritt ebenfalls bei IT-Outsourcing-Vorhaben auf, weshalb dies im strategischen Modus kritisch hinterfragt werden muss, sodass Outsourcing nicht zum Risiko wird. Da im Umstrukturierungsmodus weniger Risiken bestehen kann IT-Outsourcing einfacher durchgeführt werden und dezentrale Strukturen und Freiräume können zugelassen werden, um die Entwicklung von Innovationen zu fördern.

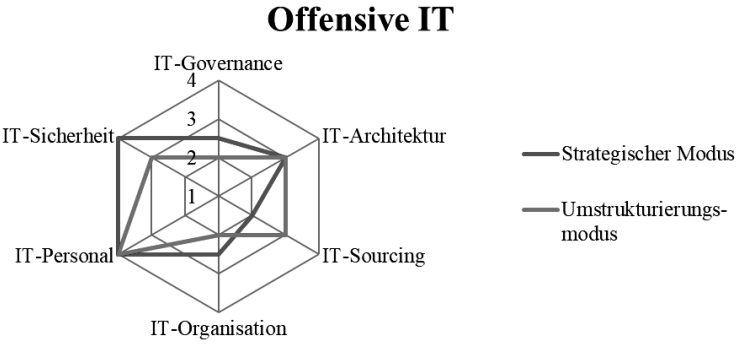


Abb. 4: Vergleich strategischer und Umstrukturierungsmodus

6 Fazit

Der Gesichtspunkt ‚aktueller Einfluss' bzw. Risiko, hat sich als ausschlaggebend herauskristallisiert. Dies gilt sowohl für die defensive als auch für die offensive IT, weshalb die vier Modi nicht auf zwei reduziert werden sollten. Es wurde versucht, in der defensiven IT zwei Rollen mit disjunktem Fokus zusammenzuführen. Den Fokus auf Kosten und die Notwendigkeit einer ununterbrochenen Verfügbarkeit und Ausfallsicherheit.

Gleichzeitig wurde versucht IT-Systeme, welche ununterbrochen verfügbar und verlässlich sein müssen mit IT-Systemen gleichzusetzten, deren Sicherheit ungeklärt ist. Doch genau dies wird im Konzept der Two-Speed-IT umgesetzt.

Es reicht folglich nicht aus zu sagen, dass eine traditionelle und eine agile IT existieren. Vielmehr müssen diese weiter detailliert werden und hinsichtlich des aktuellen Einflusses der IT zusätzlich differenziert werden. Dadurch werden die Anforderungen der IT klar ersichtlich und es wird möglich darauf zu reagieren. Hier konnte nicht nur gezeigt werden, dass eine Betrachtung der IT in vier strategischen Rollen erfolgen sollte. Gleichzeitig konnte durch die Analyse der Handlungsfelder aufgezeigt werden, welche Anforderungen jeder Modus an die Themen IT-Governance, IT-Architektur, IT-Sourcing, IT-Organisation, IT-Personal und IT-Sicherheit stellt. Legen Unternehmen also fest in welchen Modi sich ihre IT-Systeme befinden, kann daraus abgeleitet werden welcher Handlungsbedarf besteht und was dies für die IT-Strategie bedeutet. Folglich wird es Unternehmen mit der detaillierteren Unterscheidung von vier strategischen Rollen der IT möglich eine passgenauere IT-Strategie zu entwickeln, den Bedürfnissen der IT besser nachzukommen und ein gutes IT-Management aufzubauen.

Dieser Sachverhalt wurde in dieser Forschungsarbeit lediglich theoretisch ausgearbeitet. Nächste Schritte wären nun dies in der Praxis zu testen und zu verifizieren, dass mithilfe der Unterscheidung von vier strategischen Rollen der IT eine gute IT-Strategieentwicklung erfolgen kann. Des Weiteren ist es sinnvoll eine kontinuierliche Messung durchzuführen, um festzustellen wie gut die Anforderungen der Modi umgesetzt werden. So können Lücken aufgedeckt und eine bestmögliche IT-Unterstützung gewährleistet werden.

7 Literaturverzeichnis

[Al16] Alpar, P. et al.: Anwendungsorientierte Wirtschaftsinformatik. Strategische Planung, Entwicklung und Nutzung von Informationssystemen. 8. Auflage, Springer Vieweg, Wiesbaden, 2016.

[Do09] Doom, C.: An introduction to business information management. Academic and Scientific Publishers, Brüssel, 2009.

[Fo16] Foth, E.: Erfolgsfaktoren für eine digitale Zukunft. IT-Management in Zeiten der Digitalisierung und Industrie 4.0. Springer Vieweg, Berlin, 2016.

[FG07] Fröhlich, M.; Glasner, K.: IT Governance. Leitfaden für eine praxisgerechte Implementierung. Gabler Verlag / GWV Fachverlage GmbH Wiesbaden, Wiesbaden, 2007.

[GM14] Gadatsch, A.; Mayer, E.: Masterkurs IT-Controlling. Grundlagen und Praxis für IT-Controller und CIOs - Balanced Scorecard - Portfoliomanagement - Wertbeitrag der IT - Projektcontrolling - Kennzahlen - IT-Sourcing - IT-Kosten- und Leistungsrechnung. 5. Auflage, Springer Vieweg, Wiesbaden, 2014.

[GM17] Gadatsch, A.; Mangiapane, M.: IT-Sicherheit. Digitalisierung der Geschäftsprozesse und Informationssicherheit. Springer Fachmedien Wiesbaden, Wiesbaden, 2017.

[Ha09] Hanschke, I.: Strategisches Management der IT-Landschaft. Ein praktischer Leitfaden für das Enterprise Architecture Management. Hanser, München, 2009.

[Ha14] Hanschke, I.: Lean IT-Management - einfach und effektiv. Der Erfolgsfaktor für ein wirksames IT-Management. Hanser, München, 2014.

[HG15] De Haes, S.; Van Grembergen, W.: Enterprise governance of information technology. Achieving alignment and value, featuring COBIT 5. 2. Auflage, Springer, Cham, 2015.

[HS10] Hofmann, J.; Schmidt, W.: Masterkurs IT-Management. Grundlagen, Umsetzung und erfolgreiche Praxis für Studenten und Praktiker. 2. Auflage, Vieweg + Teubner, Wiesbaden, 2010.

[Jo14] Johanning, V.: IT-Strategie: Optimale Ausrichtung der IT an das Business in 7 Schritten. Springer Vieweg, Wiesbaden, 2014.

[KWW09] Keuper, F.; Wagner, B.; Wysuwa, H.: Managed Services. IT-Sourcing der nächsten Generation. Springer Fachmedien, Wiesbaden, 2009.

[KNP17] Kreutzer, R.; Neugebauer, T.; Pattloch, A.: Digital Business Leadership. Digitale Transformation - Geschäftsmodell-Innovation - agile Organisation - Change-Management. Springer Gabler, Wiesbaden, 2017.

[MB15] Mangiapane, M.; Büchler, R.: Modernes IT-Management. Methodische Kombination von IT-Strategie und IT-Reifegradmodell. Springer Vieweg, Wiesbaden, 2015.

[NM05] Nolan, R.; McFarlan, W.: Information Technology and the Board of Directors. Harvard Business Review 83, 96-106, 2005

[PS13] Pearlson, K.; Saunders, C.: Strategic Management of Information Systems. 5. Auflage, Wiley, Hoboken, NJ, 2013.

[Re13] Resch, O.: Einführung in das IT-Management. Grundlagen, Umsetzung, Best Practice. 3. Auflage, Schmidt, Berlin, 2013.

[SLT09] Saunders, M.; Lewis, P.; Thornhill, A.: Research methods for business students. 5. Auflage, Pearson Education, Harlow, 2009.

[St08] Stoll, S.: IT-Management. Betriebswirtschaftliche, ökonomische und managementorientierte Konzepte. Oldenbourg Wissenschaftsverlag GmbH, München, 2008.

[Ti17] Tiemeyer, E.: Handbuch IT-Management. Konzepte, Methoden, Lösungen und Arbeitshilfen für die Praxis. 6. Auflage, Hanser, München, 2017.

[UA16] Urbach, N.; Ahlemann, F.: IT-Management im Zeitalter der Digitalisierung. Auf dem Weg zur IT-Organisation der Zukunft. Springer Gabler, Heidelberg, 2016.

[UA17] Urbach, N.; Ahlemann, F.: Die IT-Organisation im Wandel. Implikationen der Digitalisierung für das IT-Management. HMD Praxis der Wirtschaftsinformatik 54, 300-312, 2017

[WP02] Ward, J.; Peppard, J.: Strategic Planning for Information Systems. 3. Auflage, Wiley, Chichester, 2002.

Informatik in der Anwendung

Einsatz von Netzwerksimulatoren in der Netzwerk-Lehre

Hendrik Amler[1]

Abstract: In der Netzwerklehre an Hochschulen ist der Einsatz von physikalischer Netzwerkhardware weit verbreitet. Dies beschränkt die Durchführung von Veranstaltungen auf bestimmte Räumlichkeiten und hat meist hohe Kosten zur Folge. Zudem ist die Hardware in ihrer Funktionalität beschränkt. Die Virtualisierung von Netzwerkgeräten mit einem Netzwerksimulator (NS) hebt diese Restriktion auf und kann den Weg zu neuen Lehrmethoden eröffnen. In dieser Arbeit soll evaluiert werden, welche NS für die Lehre geeignet sind und wie ein NS in eine vorhandene Lehrveranstaltung integriert werden kann. Nach einer Evaluation wird GNS3 als der geeignetste NS für den Einsatz in der Lehre bestimmt und der Einsatz in einer Lehrveranstaltung erfolgreich erprobt. Mit den gewonnenen Erkenntnissen soll zeitnah eine Nutzerstudie durchgeführt werden, um zu quantifizieren, inwieweit sich die Qualität der Lehre durch den Einsatz von Netzwerksimulatoren verbessert.

Keywords: Computernetzwerke; Virtualisierung; Lehre; GNS3

1 Motivation

Das Erlernen von praktischen Fähigkeiten im Bereich Netzwerke stützt sich in der Netzwerklehre des Fachbereichs Informatik der Hochschule Darmstadt auf das Bedienen von spezialisierter, physikalischer Hardware (Netzwerk-Appliance) in einem „face to face“[2] Lernansatz. Diese Hardware ist in der Anschaffung oft sehr teuer und veraltet schnell. Zudem fallen Kosten für Lizenzen sowie zur Freischaltung bestimmter Funktionalitäten an. Sowohl für die Studierenden als auch für Laboringenieure und Tutoren gestaltet sich der Umgang mit diesen Appliances als schwierig. Die Studierenden können die Labore ausschließlich vor Ort in speziellen Netzwerklaboren und in begrenzter Gruppenanzahl durchführen. Der ständige Wechsel der Verkabelung und inkonsistente Softwareversionen der einzelnen Geräten sind Fehlerquellen für Laboringenieure und Tutoren. Das eigentliche Ziel der Netzwerklehre, den Studierenden einen allgemeinen Wissensstand zur Erkennung und Lösung von Netzwerkproblemen unabhängig von der verwendeten Hardware zu lehren, wird oft verfehlt.

[1] Hochschule Darmstadt, Fachbereich Informatik, Haardtring 100, 64295 Darmstadt, Deutschland
hendrik.amler@h-da.de

[2] Face to face ist eine Lehrmethode, die ausschließlich auf Präsenzveranstaltungen aufbaut.

2 Ziele

Im ersten Schritt werden Projektziele definiert und priorisiert. Diese sind in Tabelle 1 aufgeführt. Hierzu wird für jeden der Stakeholder (Professor, Studierender, Laboringenieur) ein Zieldiagramm, angelehnt an Teile des KAOS-Modells nach [vL09], erstellt. Mit dieser Vorgehensweise soll sichergestellt werden, dass die Ziele aller beteiligten Stakeholder berücksichtigt werden.

Ziel	Priorisierung
Durchführung des Praktikums unabhängig von Laborausstattung	A
Kostenneutralität	A
Lizenzrechtliche Absicherung	A
Einfache und intuitive Bedienung der Software	B
Fehler im Praktikumsablauf minimieren	B
Motivation der Studierenden erhöhen	C
Nutzung studentischer Hardware	C

Tab. 1: Zielsetzung und Priorisierung

Die laborunabhängige Durchführung der Praktika hat die höchste Priorität (A). Die begrenzten Räumlichkeiten sollen entlastet und die Durchführung von Praktika in anderen Räumlichkeiten ermöglicht werden. Dies soll durch den Einsatz eines Netzwerksimulators (NS) erreicht werden. Dadurch soll eine kostenneutrale sowie lizenzrechtlich einwandfreie Lösung bewirkt werden (ebenfalls Priorität A).

Ebenso gewünscht ist eine unkomplizierte Durchführung des Praktikums durch die Studierenden (Priorität B). Komplexe Installationsschritte oder ausschließliche Konsolennutzung sollten vermieden werden. Der eingesetzte NS muss also eine intuitive und übersichtliche Oberfläche bieten.

Probleme bei der Durchführung des Praktikums sollten minimiert werden (Priorität B). Hierzu zählen z. B. Fehler in der Verkabelung der Hardware, die eine Durchführung des Praktikums behindern. Durch den Einsatz eines NS fällt die physische Verkabelung weg.

Die Ziele „Motivation der Studierenden erhöhen“ und „Nutzung studentischer Hardware“ erhalten die dritte Priorisierungsstufe C. Statt der traditionellen „face to face“-Lehrmethode soll ein Blended Learning Ansatz verfolgt werden, der Präsenzstunden in der Hochschule sowie E-Learning Methoden verbindet. Ferner sollen weitere Open Educational Resources (OERs) wie z. B. Moodle verwendet werden, um den Lernprozess zu unterstützen.

Das Erreichen der definierten Ziele soll dem übergeordnete Ziel „Verbesserung der Qualität der Lehre“ dienen.

3 Vorgehensweise

Um die gesetzten Ziele durch Einsatz eines NS zu erreichen, müssen folgende Schritte durchgeführt werden: 1. Auswahl eines geeigneten NS, 2. Modellierung der Praktika innerhalb dieses NS, 3. Ausrollen der Software in den Laboren sowie 4. Durchführung der Praktika mit NS.

Hinsichtlich des positiven Einflusses auf den Lerneffekt der Studierenden durch den Einsatz eines NS in der Netzwerklehre wird eine Studie von Gil et al. [Gi14] zugrunde gelegt. Gil et al. hat diese positiven Effekte insbesondere bei Studierenden mit geringen Kenntnissen im Bereich Computernetzwerke und TCP/IP festgestellt. In [Gi13] wurden mehrere Tests basierend auf statistischen Korrelationen sowie Regressionsanalysen durchgeführt, um die Beziehung zwischen der von den Studierenden erreichten Note und der Anzahl und Art der im Lernprozess verwendeten OERs wie z. B. Netzwerksimulatoren nachzuweisen. Es konnte eine signifikante Verbesserung der Noten mit steigender Anzahl von OERs beobachtet werden.

3.1 Auswahl des Netzwerksimulators

Für das Definieren von Anforderungen für einen NS zum Einsatz in der Lehre wird auf die folgenden Kriterien von Pizzonia und Rimondini [PR16] zurückgegriffen:

Installationsaufwand: Die Studierenden sollten ihre Zeit mit dem Konfigurieren von Netzwerken statt mit der Installation des NS verbringen.

Benutzbarkeit: Das Erlernen der Grundfunktionalitäten des NS sollte möglichst wenig Zeit kosten.

Genauigkeit: Der NS sollte eine physikalische Appliance in Bezug auf Bedienbarkeit und Verhalten möglichst wirklichkeitsgetreu nachbilden können.

Anmerkung: Dieser Aspekt wird in diesem Vergleich von geeigneten NS nicht berücksichtigt, da die realitätsnahe Nachbildung eines Netzwerks für diesen Einsatzzweck nur eine untergeordnete Rolle spielt.

Teilen von virtuellen Laboren: Die Studierenden sollen in den Laboren zum Teil vorkonfigurierte Netzwerke bearbeiten. Daher ist eine Export- bzw. Importfunktion von Laboren nötig.

Niedrige Hardwareanforderungen: Die Emulation von komplexen Netzwerktopologien soll auf mittelpreisiger Desktop- bzw. Notebookhardware möglich sein.

Anmerkung: Auch diese Anforderung wird hier vernachlässigt, da alle NS im hier anzustellenden Vergleich mit den im Praktikum verwendeten Topologien auf mittelpreisiger Laptophardware lauffähig sind. Es gibt einige Einschränkungen, die in Abschnitt 4 genannt werden.

Unterstützung von vielen Netzwerktechnologien: Der NS sollte gängige Netzwerktechnologien nativ unterstützen sowie Unterstützung für weitere Technologien bieten.

Mit Ausnahme von „Genauigkeit“ und „Niedrige Hardwareanforderungen“ werden die von Pizzonia und Rimondini [PR16] genannten Anforderungen für den anzustellen Vergleich von NS übernommen und folgende Punkte hinzugefügt:

Multiplatform-Support (MP-Support): MP-Support ist ein untergeordnetes Ziel der Nutzung studentischer Hardware. Der NS sollte mit allen gängigen Betriebssystemen funktionieren. Dies ist essentiell, um den Studierenden die Bearbeitung der Aufgaben mit eigener Hardware zu ermöglichen.

Niedrige Lizenzkosten: Die Lizenzkosten sollten minimal sein und das vorher definierte Ziel der Kostenneutralität erfüllen.

Im Weiteren wird eine Auswahl von Netzwerksimulatoren anhand der genannten Anforderungen miteinander verglichen. Ein Vergleich der einzelnen NS ist in Tabelle 2 abgebildet und dient als Entscheidungshilfe. Die Auswahl orientiert sich an Mohtasin et al. [Mo16] sowie Pizzonia et. al. [PR16]. Ferner wurde der Cisco Packet Tracer mit in den Vergleich aufgenommen, da dieser den Studierenden vereinzelt von Laboringenieuren zur Vor- bzw. Nachbereitung von Laboren empfohlen wurde.

NS	Benutzbarkeit	MP-Support	Installationsaufwand	Lizenzkosten	Netzwerktechnologien	Teilen von Laboren
GNS3[3]	Gut	Ja	Mittel	Entfällt	Viele	Ja
Mininet[4]	Schwerfällig	Ja	Hoch	Entfällt	Viele	Ja
Cisco Packet Tracer[5]	Gut	Ja	Gering	Entfällt	Cisco[6]	Ja
Cisco Modeling Labs VIRL[7]	Gut	Nein	k.A.	$200/a/user	Cisco	Ja
Netsim[8]	Ausreichend	Nein	k.A.	$179/a/user	Cisco	Ja

Tab. 2: Übersicht ausgewählter Netzwerksimulatoren

Die Metrik „Benutzbarkeit“ ist eine subjektive Einschätzung der für die Praktika notwendigen Bedienelemente des NS. Der Installationsaufwand richtet sich nach der aufzuwendenden Zeit zur Installation des NS inklusive der Benutzeroberfläche. Die Anzahl der Netzwerktechnologien richtet sich nach den bisher in verschiedenen Praktika genutzten

[3] `https://www.gns3.com` (Letzter Zugriff: 25.03.2018)
[4] `http://mininet.org/` (Letzter Zugriff: 09.04.2018)
[5] `https://www.netacad.com/courses/intro-packet-tracer/` (Letzter Zugriff: 09.04.2018)
[6] Beschränkte Funktionalität
[7] `https://www.cisco.com/c/en/us/products/cloud-systems-management/modeling-labs/index.html` (Letzter Zugriff: 09.04.2018)
[8] `http://www.boson.com/netsim-cisco-network-simulator` (Letzter Zugriff: 25.04.2018)

Netzwerktechnologien. Die weiteren Metriken wurden den Datenblättern der NS entnommen.

Aufgrund der GPLv3-Lizenzierung und somit kostenlosen Nutzung, dem benutzerfreundlichen GUI sowie einer Vielfalt von Features fiel die Wahl auf den NS *GNS3*. Das GNS3-GUI ist in Abbildung 1 zu sehen. Mininet hingegen bietet selbst keine Benutzeroberfläche. Das optional installierbare GUI ist aufgrund der schwerfälligen Benutzbarkeit nicht für den Einsatz in der Lehre zu empfehlen. Der Cisco Packet Tracer bietet nicht alle benötigten Netzwerktechnologien. Die weiteren proprietären Lösungen erfüllen nicht das definierte Ziel der Kostenneutralität.

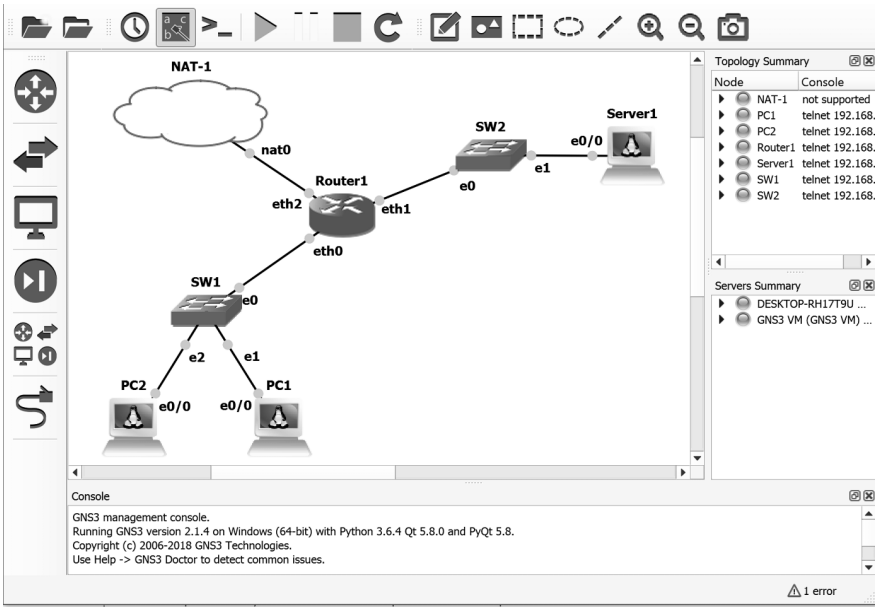


Abb. 1: GNS3 GUI

3.2 Erstellung von Praktika mit GNS3

Nach der Auswahl des NS werden die Praktika in GNS3 erstellt. Diese sind die Grundlage für die zu bearbeitenden Aufgaben der Studierenden. Die Praktika werden im Rahmen der Wahlpflichtveranstaltung *Internetworking* [Mo17] für Studierende im 5. Semester erstellt. GNS3 bietet die Möglichkeit, eine Vielzahl von Netzwerk-Appliances zu virtualisieren. Diese können in einem GUI beliebig miteinander verbunden und zu einem Netzwerk geformt werden. Der Datenverkehr jedes einzelnen Pfades kann mit dem Packet-Capturer Wireshark untersucht werden. Ferner ist es möglich, das Erstellen von Netzwerken über

eine REST-API zu automatisieren. Für das Erstellen der Praktika ist besonders die Export-Funktion von Projekten hilfreich, die eine Bearbeitung auch außerhalb des Labors und auf studentischer Hardware ermöglicht. Aus lizenzrechtlichen Gründen wurden ausschließlich Open-Source Appliances verwendet. Die Aufgabenstellungen der Praktika sowie die in Abschnitt 3.4 beschriebene Durchführung sind nach folgenden Lernzielen konzipiert:

Die Studierenden:

1. können den Aufbau und die erweiterten Funktionen von Internet-Protokoll-basierten Netzen analysieren,
2. können die erworbenen Kenntnisse über Aufbau und Funktion anwenden, um selbständig IP-basierte Netzwerke und Dienste zu konstruieren, aufzubauen und zu betreiben und
3. kennen die weiterführenden Literaturquellen und verstehen die Vorgehensweisen, um ihr Wissen an den schnellen Wandel im Umfeld der Datennetzwerke anzupassen.

3.3 Installation und Benutzung von GNS3

Die Installation auf den Netzwerklabor-PCs erfolgt automatisiert über GoSA[9]. GoSA rollt auf den Labor-PCs ein Linux System über ein Preboot Execution Environment (PXE) aus. Dieses führt eine vollständige Installation eines Ubuntu Xenial Betriebssystem über ein hochschulinternes Repository durch. Zusätzliche Software wie z. B. GNS3 und Abhängigkeiten werden über Skripte nachinstalliert. GNS3 verwendet ein eigenes externes Repository, welches ebenfalls automatisiert ins System eingepflegt werden muss.

Die Projektdateien für die Labore werden den Studierenden über die hochschulinterne Nextcloud bzw. Moodle zur Verfügung gestellt. Die Projektdateien weisen eine Größe von 10 MB bis maximal 250 MB auf und enthalten alle zum Bearbeiten des Labors notwendigen Konfigurationen sowie Images der Netzwerkinstanzen. Diese können auf einer aktuellen GNS3-Version problemlos importiert werden.

3.4 Durchführung von Praktika mit GNS3

In den Praktika werden den Studierenden teilweise unvollständige Netzwerke gegeben, die diese dann eigenständig analysieren, vervollständigen und nach gewissen Vorgaben (z. B. Zone A darf mit Zone B kommunizieren) konfigurieren müssen. Die Netzwerke orientieren sich an realen Netzen, wie sie z. B. in kleinen Unternehmen zum Einsatz kommen. Eine Beispieltopologie ist in Abbildung 2 gegeben. Nachdem die Netze vollständig konfiguriert sind, sind die Studierenden angehalten, Datenverkehr zu erzeugen (z. B. DoS Traffic) und

[9] `https://oss.gonicus.de/labs/gosa/` (Letzter Zugriff: 12.04.2018)

Messungen an vordefinierten Messpunkten vorzunehmen. Im Anschluss werten die Studierenden die Messergebnisse in Gesprächen mit dem Lehrbeauftragten aus und bewerten die eigene Konfiguration unter anderem nach Sicherheitskriterien. In einigen Praktika wird wissenschaftliche Literatur herangezogen, um Netzwerkprotokolle hinsichtlich der verschiedenen Sicherheitsziele von Netzwerkprotokollen wie TLS und QUIC miteinander zu vergleichen [G7] [Ly15]. Durch diese Herangehensweise werden die in Abschnitt 3.2 unter Ziffer 1 bis 3 genannten Kompetenzen gestärkt.

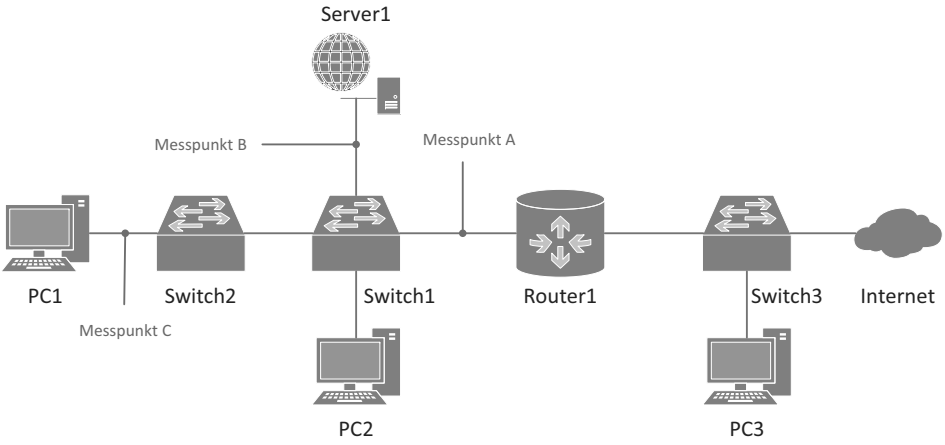


Abb. 2: Beispieltopologie für ein Praktikum

4 Erste Erkenntnisse

Erste Erfahrungen nach Durchführung eines Praktikums mit dem NS *GNS3* geben erste Hinweise auf die Erfüllbarkeit der unter Abschnitt 2 formulierten Ziele.

Durchführung des Praktikums unabhängig von Laborausstattung

Das Ausrollen von GNS3 im Labor führte zu einigen Problemen bei den Dateisystemrechten der Studierenden. Komponenten wie Wireshark (Packet Capture) oder uBridge[10] benötigen root-Rechte auf dem Labor-PC. Da die Nutzer ihre Rechte über ein LDAP-System und nicht über lokale Gruppen bekommen, werden die benötigten Rechte systemweit über Skripte den einzelnen Applikationen zugewiesen. Die Dockerkomponente von GNS3 benötigt ebenfalls root-Rechte. Ein Zuweisen von root-Rechten für das Erstellen von Docker-Instanzen für alle Nutzer würde die Rechtevergabe umgehen, weshalb die Dockerfunktionalität von GNS3 vorerst nicht im Labor genutzt werden kann.

[10] uBridge erstellt user-land bridges zwischen verschiedenen von GNS3 verwendeten Technologien z. B. Ethernet und TAP-Interfaces

Lizenzrechtliche Absicherung

Es sollte auf eine korrekte Lizenzierung der verwendeten NS und der virtuellen Netzwerkappliances geachtet werden. Eine Anfrage bei Cisco zur Verwendung von Cisco Images in z. B. GNS3 blieb leider erfolglos. Die Frage nach einer Lizenzierung für das Cisco Modeling Lab bzw. VIRL blieb unbeantwortet. Die Verwendung von GNS3 und Open Source Netzwerkappliances wie VyOS[11] oder Open vSwitch[12] ist lizenzrechtlich nicht zu beanstanden.

Einfache und intuitive Bedienung der Software

Auf der Hardware der Studierenden verursachte besonders die Cloud / NAT Instanz von GNS3, die eine Schnittstelle zu anderen Netzwerken wie z. B. dem Internet bietet, Probleme auf Windows und MacOS. Es sollten möglichst Praktika ohne Cloud / NAT erstellt werden bis die Fehler behoben sind. Auf den Linux Labor-PCs konnten die Probleme nicht beobachtet werden. Ferner sollte die Hardware der Studierenden Virtualisierungssupport über Intel VT-x[13] bzw. AMD-V[14] bieten.

Nutzung studentischer Hardware

Die Installation bereitete drei der 12 Studierenden Schwierigkeiten. Für zwei Studierende war die Konfiguration der für GNS3 notwendigen virtuellen Maschine (GNS3-VM) unter MacOS / Windows eine Herausforderung. Bei einem Studierenden konnten keine Netzwerkpakete mitgeschnitten werden. Dies konnte auf einen Versionskonflikt zwischen GNS3 und bereits installierten Abhängigkeiten zurückgeführt werden. Weiterhin spielt die Wahl der Virtualisierungsumgebung eine entscheidende Rolle für die Performance, die von 3 Studierenden bemängelt wurde. Virtualbox z. B. bietet keine Unterstützung für Nested Virtualization und hat auf leistungsschwacher Hardware Reaktionszeiten von einigen Sekunden beim Absetzen eines Befehls auf einer Netzwerkinstanz zur Folge. Es wird ausdrücklich der Einsatz des VMWare Players als Virtualisierungsumgebung für die GNS3-VM empfohlen. Unter Linux ist die GNS3-VM nicht erforderlich, da hier KVM[15] nativ ausgeführt werden kann.

[11] `https://vyos.io/` (Letzter Zugriff: 12.04.2018)
[12] `https://www.openvswitch.org/` (Letzter Zugriff: 12.04.2018)
[13] `https://www.intel.com/content/www/us/en/virtualization/virtualization-technology/intel-virtualization-technology.html` (Letzter Zugriff: 08.04.2018)
[14] `https://www.amd.com/en/technologies/virtualization` (Letzter Zugriff: 08.04.2018)
[15] `https://www.linux-kvm.org/` (Letzter Zugriff: 30.04.2018)

Interview

Zum Thema „Einsatz von GNS3 in der Lehrveranstaltung Internetworking“ wurde ein Interview mit einen Studierenden geführt (Persönliche Kommunikation, 24.04.2018). Das Interview wird nach dem „General Interview Guide Approach“ absolviert [TI10]. Im Gegensatz zu einer informellen Konversation bietet diese Interviewmethode einen strukturierten Rahmen, der aber Flexibilität bei der Durchführung des Interviews zulässt. So können z. B. Fragen während des Interviews umformuliert oder je nach Antwort des Befragten angepasst werden. Der Fragebogen für das Interview ist in Abbildung 3 im Appendix abgebildet. Nach Selbsteinschätzung der eigenen Studienleistungen ordnet sich der Befragte leicht über dem Durchschnitt im Vergleich zu Kommilitonen ein. Die gestellten Fragen im Interview sollen sowohl das Erreichen von den in Abschnitt 3.2 genannten Lernzielen des Moduls *Internetworking*, als auch die für Studierende wichtigen Projektziele aus Abschnitt 2 überprüfen. Ferner sollten Vorteile hervorgehoben und etwaige Probleme beim Einsatz des Netzwerksimulators eingegrenzt werden.

Der allgemeine Eindruck des Befragten zum Praktikum war sehr positiv. Der Befragte ist davon überzeugt, dass das eigene Netzwerkverständnis durch den Einsatz von GNS3 wesentlich verbessert wurde. Die Möglichkeit, die Praktika zu Hause vorzubereiten, wurde positiv hervorgehoben. Dies hat das Experimentieren und Explorieren neuer Funktionen von GNS3 gefördert. Ein zeitintensives Vereinbaren zusätzlicher Termine zur Bearbeitung der Aufgaben bei nahezu voller Auslastung des Labors entfällt. Zudem sinkt der Zeit- und Leistungsdruck im Labor bei Bearbeitung der Aufgaben zu Hause. Die Installation von GNS3 auf dem eigenen PC bereitete keine Probleme. Die Software hat allerdings nicht auf Anhieb funktioniert. Die komplexe Konfiguration von GNS3 wie z. B. die Einrichtung der GNS3-VM unter Windows hat zu viel Zeit gekostet. Hierdurch wurde der Zeitaufwand für die Vor- bzw. Nachbereitung als nicht angemessen bewertet.

Nachdem nun ein Basiswissen besteht, traut sich der Befragte auch in Zukunft zu, mit GNS3 zu arbeiten. Die Benutzbarkeit des Netzwerksimulators GNS3 wurde als sehr gut bewertet. Die Bedienung der Benutzeroberfläche bereitete keine Probleme. Einzig die Bedienung der Netzwerkinstanz Cloud war nicht auf den ersten Blick ersichtlich aufgrund unverständlicher Einstellungsmöglichkeiten. Es sei nach seiner Aussage vergleichsweise einfach, eine Netzwerktopologie mit unterschiedlichen Netzwerkappliances aufzusetzen und zu konfigurieren. Die Performance von GNS3 auf dem eigenen Laptop wurde als mittelmäßig eingestuft. Die Verwendung von VmWare Player statt Virtualbox brachte einen spürbaren Performanceschub.

Bei der Frage nach einer Präferenz in der Methodik der Praktika für die Veranstaltungen *Internetworking* und *Netzwerke* äußerte sich der Befragte positiv für den Einsatz von GNS3 in höheren Semestern. Für den Einsatz in Grundlagenveranstaltungen der niederen Semester äußerte der Befragte allerdings Bedenken aufgrund der erfahrenen Konfigurationsprobleme.

Das Interview wurde vorerst nur mit einer Testperson durchgeführt. Die Aussagen sind

daher nicht repräsentativ und das Interview sollte mit weiteren Studierenden wiederholt werden.

5 Zusammenfassung und Ausblick

Die in Abschnitt 2 gesetzten Ziele werden erreicht. Es wurde eine Evaluierung von verschiedenen NS durchgeführt und GNS3 für den Einsatz in der Lehre ausgewählt. Die Praktika können mit dieser Lösung außerhalb des Labors stattfinden und entlasten so die begrenzten Räumlichkeiten. Den Studierenden ist die Vor- bzw. Nachbereitung zu Hause möglich, die Anwesenheitszeit im Labor kann für gezielte Fragen und Hilfestellungen genutzt werden. Da ausschließlich Open Source Software zum Einsatz kommt, müssen keine kostspieligen Lizenzen angeschafft werden. Die entworfenen Praktika wurden in einer Lehrveranstaltung erprobt und bieten erste Erfahrungen und Eindrücke der Studierenden. Das Interview mit einem Studierenden brachte weitere wertvolle Erkenntnisse wie z.B. eine erhöhte Motivation durch die Möglichkeit, die Praktika zu Hause zu bearbeiten. Anfängliche Probleme wie z. B. Installationsschwierigkeiten unter einigen Betriebssystemen, die in Abschnitt 4 beschrieben werden, sollen durch eine bereits entworfene, verbesserte Version der Praktika gemildert werden. In den folgenden Iterationen wird die Durchführung der Praktika weiter optimiert.

Im nächsten Schritt soll über die ersten Erkenntnisse hinaus evaluiert werden, inwieweit der Einsatz eines NS die Lehre verbessert. Hierfür sollen Methoden für die Quantifizierung der Ergebnisse ausgewählt werden. Dies soll durch eine Nutzerstudie für die Gruppen Studierende, Lehrende sowie Laboringenieure unterstützt werden. Die Nutzerstudie soll im Rahmen der Grundlagenveranstaltung *Netzwerke* [Mo17] durchgeführt werden. Der Einsatz von GNS3 in dieser Veranstaltung soll in zwei Testgruppen erprobt werden. Als Vergleich werden zwei Referenzgruppen herangezogen, welche die Veranstaltung in der herkömmlichen Lehrmethode ohne den Einsatz von GNS3 durchführen.

Literaturverzeichnis

[Gi13] Gil, Pablo; Candelas-Herías, Francisco A; Jara, Carlos A; García, Gabriel J; Torres, Fernando: Web-based OERs in computer networks. International Journal of Engineering Education, 29(6):1537–1550, 2013.

[Gi14] Gil, Pablo; Garcia, Gabriel J; Delgado, Angel; Medina, Rosa M; Calderon, Antonio; Marti, Patricia: Computer networks virtualization with GNS3: Evaluating a solution to optimize resources and achieve a distance learning. In: Frontiers in Education Conference (FIE), 2014 IEEE. IEEE, S. 1–4, 2014.

[G7] Günther, Felix; Hale, Britta; Jager, Tibor; Lauer, Sebastian: 0-RTT Key Exchange with Full Forward Secrecy. In (Coron, Jean-Sébastien; Nielsen, Jesper BuusEditors, Hrsg.): Advances in Cryptology – EUROCRYPT 2017, Jgg. 10212. Springer International Publishing, S. 519–548, 2017.

[Ly15] Lychev, Robert; Jero, Samuel; Boldyreva, Alexandra; Nita-Rotaru, Cristina: How secure and quick is QUIC? Provable security and performance analyses. In: Security and Privacy (SP), 2015 IEEE Symposium on. IEEE, S. 214–231, 2015.

[Mo16] Mohtasin, R; Prasad, PWC; Alsadoon, Abeer; Zajko, G; Elchouemi, A; Singh, Ashutosh Kumar: Development of a virtualized networking lab using GNS3 and VMware workstation. In: Wireless Communications, Signal Processing and Networking (WiSPNET), International Conference on. IEEE, S. 603–609, 2016.

[Mo17] Modulhandbuch des Studiengangs Informatik Bachelor. Hochschule Darmstadt, Fachbereich Informatik, 2017.

[PR16] Pizzonia, Maurizio; Rimondini, Massimo: Netkit: network emulation for education: NETKIT: NETWORK EMULATION FOR EDUCATION. Software: Practice and Experience, 46(2):133–165, Feb 2016.

[TI10] Turner III, Daniel W: Qualitative interview design: A practical guide for novice investigators. The qualitative report, 15(3):754, 2010.

[vL09] van Lamsweerde, Axel: Requirements Engineering: From System Goals to UML Models to Software Specifications. Wiley Publishing, 1st. Auflage, 2009.

Fragestellung	**Ziel**
Haben Sie die Praktika für das Modul *Internetworking* außerhalb des Labors bearbeitet?	Voraussetzung für das Interview
Haben Sie das Modul *Netzwerke* besucht?	Voraussetzung für das Interview
Hatten Sie Schwierigkeiten bei der Installation der Netzwerksimulationssoftware GNS3?	Wird das Projektziel „Bearbeitung von Aufgaben außerhalb des Labors“ erfüllt?
Wie viele Stunden haben Sie durchschnittlich für die Vor- bzw. Nachbereitung aufgebracht? Fanden Sie dies angemessen?	Wird durch den Einsatz von GNS3 ein erheblicher Mehraufwand für Studierende erzeugt?
Wo gab es Zeiteinsparungen bzw. zusätzliche Zeitaufwände bei der Nutzung der Software?	Eingrenzen von Mehraufwand bzw. Hervorheben von Vorteilen der Software
Wie bewerten Sie die Benutzbarkeit von GNS3?	Erfüllt GNS3 das Projektziel „Einfache und intuitive Benutzung der Software“?
Gab es Vorteile, die bei der Benutzung der Software aufgefallen sind?	Hervorheben von Vorteilen der Software
Hatten Sie Schwierigkeiten bei der Benutzung von GNS3?	Eingrenzung potentieller Problemfelder der Software
Wie bewerten Sie die Performance von GNS3?	Wird das Projektziel „Nutzung auf studentischer Hardware“ erfüllt?
Wie bewerten Sie Ihre eigenen Studienleistungen im Vergleich zu Ihren Kommilitonen?	Eigene Einschätzung der Leistungen
Haben Sie das Gefühl, durch den Einsatz von GNS3 ein besseres Verständnis über den Aufbau und die Funktionsweise von Netzwerken bekommen zu haben?	Erreichen von Lernzielen des Moduls *Internetworking*
Trauen Sie sich nach Abschluss der Veranstaltung zu, weitere Netzwerktopologien mit dem Netzwerksimulator GNS3 zu bearbeiten?	Erreichen von Lernzielen des Moduls *Internetworking*
Haben Sie den Netzwerksimulator bereits außerhalb der Veranstaltung Internetworking z.B. für eigene Projekte verwendet?	Erreichen von Lernzielen des Moduls *Internetworking*
Vergleichen Sie die Praktika der Veranstaltungen *Netzwerke* sowie *Internetworking* im Bezug auf den Einsatz der Netzwerkvirtualisierungssoftware GNS3 miteinander. Welche Methodik der Durchführung der Praktika präferieren Sie?	Eignung von GNS3 in der Netzwerklehre

Abb. 3: Fragestellungen und Ziele des Interviews

Nutzung von Robotern im Informatikunterricht – ein Lösungsvorschlag

Lina Peters[1] Nick Fahrendorff[2] Dennis Debeye[3] Dennis Alt[4]

Abstract: Die heutige digitalisierte Welt verlangt die Vermittlung von MINT-Kompetenzen. Gute Bildung in dem Bereich der Informatik ist also unverzichtbar. Daher ist es wichtig Schülerinnen und Schüler für die Themen der MINT-Fächer zu motivieren. Im Schulalltag finden Lehrer mit den ihnen zur Verfügung stehenden, begrenzten Mitteln kaum Möglichkeiten den Informatikunterricht spannend und motivierend zu gestalten. Genau dieser Situation soll die vorliegende Arbeit entgegenwirken, indem sie mit Robotern eine motivierende Alternative für Lehrinhalte der Informatik thematisiert. Anhand der Betrachtung der aktuellen Lehrplaninhalte, verfügbarer Roboter und angebotener Weiterbildungsmöglichkeiten für das Lehrpersonal kommen die Autoren zu dem Schluss, dass der Einsatz von Robotern den Informatikunterricht an deutschen Schulen nachhaltig verbessern kann.

Keywords: Informatikunterricht, Schulroboter, Service Learning, MINT, LEGO Mindstorms, Roberta

1 Einleitung

Die Digitalisierung in der Arbeitswelt ist ein unaufhaltsamer Prozess, der zunehmend nach Fachkräften in den Bereichen Softwareentwicklung, Programmierung und IT-Anwenderberatung verlangt, während Berufsbilder in anderen Branchen aussterben oder den sicheren Umgang mit digitalen Arbeitsmitteln erfordern. Sowohl die Bundesagentur für Arbeit als auch der Bundesverband Informationswirtschaft, Telekommunikation und neue Medien (bitkom) attestieren Deutschland jedoch gerade im IT-Bereich einen Fachkräftemangel [17b], [Be17a]. Gleichzeitig verzeichneten die deutschen Hochschulen im Wintersemester 2017/2018 sogar einen Rückgang an Studienanfängern im Fach Informatik [17d]. Die Erklärung für dieses Missverhältnis ist unter anderem an den Schulen zu suchen: Der Informatikunterricht in Deutschland steckt in einer tiefen Krise. Es fehlt an Lehrern und zeitgemäßer Ausstattung [17a, S.82ff.]. [17e, S.10]. Das Bildungssystem läuft Gefahr eine Generation digitaler Analphabeten heranzuziehen, die uns wirtschaftlich weit hinter andere führende Industrienationen zurückfallen lassen könnte. Jedoch sind möglicherweise *Mensch und Material* nicht die einzigen Faktoren, die in dieser Krise eine Rolle spielen: Betrachtet man Beispielhaft Nordrhein-Westfalen – mit einer Zahl von fast zwei Millionen,

[1] HTWK Leipzig , IMN, Karl-Liebknecht-Str. 132 , 04277 Leipzig lina.peters@stud.htwk-leipzig.de
[2] HTWK Leipzig , IMN, Karl-Liebknecht-Str. 132 , 04277 Leipzig nick.fahrendorff@stud.htwk-leipzig.de
[3] HTWK Leipzig , MIM, Karl-Liebknecht-Str. 132 , 04277 Leipzig dennis.debeye@stud.htwk-leipzig.de
[4] HTWK Leipzig , IMN, Karl-Liebknecht-Str. 132 , 04277 Leipzig dennis.alt@stud.htwk-leipzig.de

das Bundesland mit den meisten Schülern an allgemeinbildenden Schulen – so fällt auf, dass die Teilnehmerzahlen im Wahlpflichtfach Informatik deutlich hinter denen der anderen MINT-Fächern zurückliegen (im Schuljahr 2016/17 insgesamt 41.318 Teilnehmer in Informatik-Grundkursen der gymnasialen Oberstufe; auf dem vorletzten Platz liegt Physik mit 78.590 Teilnehmern). Lediglich 10.762 Schüler in Nordrhein-Westfahlen, 13%, belegen Informatik noch im letzten Schuljahr [17c]. Wir vermuten aus eigener Erfahrung die folgenden Ursachen: Die curricularen Fragestellungen der Informatik sind für die Schülerinnen und Schüler nicht ansprechend. Anders als in anderen Naturwissenschaften mangelt es dem Informatikunterricht an praktischen Anwendungsbeispielen. Bei Programmieraufgaben im klassischen Informatikunterricht haben Schülerinnen und Schüler – mangels individueller Rückkopplung – kaum Gelegenheit ihre eigene Leistung einzuschätzen. Die in [JTA15] beschriebene, im Regierungsbezirk Münster durchgeführte Studie zeigt tatsächlich, dass u.a. das Desinteresse an Computern und interessantere Alternativen im Wahlpflichtbereich dazu führen, dass in der Sekundarstufe I Informatik nicht angewählt wird.
Eine Möglichkeit den Informatikunterricht attraktiver und motivierender zu gestalten ist der Einsatz von Robotern als Lernobjekt im Unterricht, wie u.a. in [SG09], [ÇDY17] und [Ch15] beobachtet werden konnte. Auch im persönlichen Gespräch mit Schülerinnen und Schülern, die bereits heute mit Robotern beschult werden, zeigte sich eine gesteigerte Motivation den Inhalten und der Programmierung gegenüber. Daher soll der Einsatz von Robotern im Unterricht unter Einhaltung der bestehenden Curricula geprüft werden.
In der vorliegenden Arbeit vergleichen wir zunächst die Lehrpläne aller deutschen Bundesländer. In einem zweiten Schritt geben wir einen Überblick über die verschiedenen Lehr- und Lernroboter. Weiterhin betrachten wir die zur Verfügung stehenden Lehrmittel und Schulungen für Lehrerinnen und Lehrer. In der anschließenden Auswertung wird analysiert, wie gut sich die zuvor generalisierten Anforderungen der Curricula durch die unterschiedlichen vorgestellten Roboter abbilden lassen und somit, ob diese ein geeignetes Lehr- und Lernmittel für den Informatikunterricht in Deutschland darstellen.

2 Stand der Wissenschaft

[SS15], [Be17b] und [De17] untersuchen aktuelle Bildungseinrichtungen in Hinblick auf das Thema digitale Bildung. Dabei kommen sie alle zu dem Ergebnis, dass momentane Bildungsmöglichkeiten zu gering sind. Sie werden sogar als „lückenhaft“ und „uneinheitlich“ [SS15, S.30] bezeichnet. Bemängelt wird, dass es an zeitgemäßer technischer Ausstattung fehlt [De17, S.7] und dass die Grundlage für eine gute Bildung im technischen Bereich abhängig von der Wahl der Schule und den Interessen der Schüler und Schülerinnen ist, so [SS15, S.30] nach einer Untersuchung der Umsetzung von Kompetenzen im Lehrplan und Unterricht. Ein Blick in die Zukunft verspricht hier keine Verbesserungen. So entscheiden sich nicht ausreichend Schüler oder Schülerinnen für einen Beruf oder ein Studium im MINT-Bereich [De17, S.7]. Der Leistungskurs Informatik ist deutschlandweit unterrepräsentiert [De17] und es mangelt stark an Lehrer-Nachwuchs in diesem Themengebiet [De17, S.7]. Daher ist es von besonderer Wichtigkeit vorhandenes Lehrpersonal durch Schulungen

zusätzliche Kompetenzen anzueignen. Dieses Themas nimmt sich [Be17b] an und zieht hier das Fazit, dass die Weiterbildungsmöglichkeiten sehr gering sind und nur von motivierten Lehrern wahrgenommen würden. Einen Ausweg sieht [De17] hier in digitalen Technologien, welche als „große Chance zur Verbesserung der Lehr- und Lernwelten an Schulen und Hochschulen“ [De17, S.4] angesehen werden. Diesen Ansatz verfolgt auch [Dü17]. Hierin wird geprüft, inwiefern sich verschiedene Roboter für den Einsatz im Unterricht eignen. Dazu werden verschiedene Modelle miteinander verglichen. Kriterien für die Eignung seien hierbei ein geringer Preis, ein geringer Platzbedarf sowie die Programmierbarkeit. Den Autoren dieser Arbeit sind diese Kriterien jedoch nicht ausreichend. Es sollte auch geprüft werden, inwiefern sich Roboter und die Robotik im Allgemeinen dazu eignen in den Lehrplan der Informatik integriert zu werden und diesen bestmöglich umzusetzen.

3 Lehrpläne

Die Lehrpläne für das Fach Informatik unterscheiden sich in Deutschland zwischen den Bundesländern stark, wobei einige große Themenblöcke jedoch auch über Ländergrenzen hinweg wiederzufinden sind. Informatik wird zumeist in der Mittel- und Oberstufe unterrichtet. Wir wollen uns in dieser Arbeit jedoch vor allem auf die gymnasiale Oberstufe beschränken. Hier sehen die Lehrpläne vor allem folgende Themenschwerpunkte für den Informatikunterricht vor:

Rechnernetze und Kommunikation Innerhalb dieses Themenblocks divergieren die Bundesländer stark. Er enthält bundesweit Themen wie *Internetprotokoll* [He16], Schichtenmodell und Client-Server-Lösung [Se09] oder auch *die Kommunikation zwischen Mensch und Maschine* [Sä11]. Oft werden die Themen Rechnernetze und Kommunikation als Bestandteil des Themenbereichs *Informatiksysteme* behandelt. [Mi14] Hiermit sind oftmals der Aufbau eines Rechners und die von-Neumann-Architektur verbunden. [La06]

Algorithmen und Datenstrukturen Algorithmen sollen meist am Beispiel von Sortier- und Such-Algorithmen und deren Effizienzbetrachtung gelehrt werden. Des Weiteren sollen die Schüler sich mit einfachen Datentypen und Datenstrukturen auseinandersetzen. [Sa06] [Rh11] Allgemeiner ist in vielen Lehrplänen der Entwurf, die Analyse und die Implementierung von Algorithmen, teilweise Rekursion, vorgesehen [He16]. Für den Leistungskurs findet man in einigen Lehrplänen auch Themen wie Backtracking, *Divide & Conquer* und Vererbung. [Mi02]

Daten und Datenbanken Die Schüler sollen lernen Daten zu modellieren und zugehörige Operationen auf die Daten anzuwenden. Primär geht es bundesweit darum Informationen unterschiedlichster Form zu strukturieren und zu speichern. [Th12] [Mi11]

Sprachen und Automaten Oftmals sollen die Begriffe Syntax und Semantik [Se10] sowie der Unterschied zwischen formalen und natürlichen Sprachen erklärt werden [Fr09]. Des Weiteren werden in einigen Lehrplänen Grammatiken regulärer und kontextfreier

Sprachen [Mi14] oder auch die Chomsky-Hierarchie, für den Leistungskurs, aufgeführt. Im Grundkurs werden meistens endliche Automaten thematisiert, während die Lehrpläne für den Leistungskurs zusätzlich Nicht-Determinismus vorsehen. [He16]

Softwaretechnik Ein größeres Projekt über einen längeren Zeitraum ist ebenfalls häufig vorgesehen. Dadurch sollen Kompetenzen der Softwareentwicklung, wie das Dokumentieren und Testen, vermittelt werden. [Th12] [He16]

Information, Mensch und Gesellschaft Hier werden in den verschiedenen Bundesländern sehr unterschiedliche Schwerpunkte gesetzt. Oftmals sollen die Schüler Themen wie Datenschutz und Datensicherheit vermittelt bekommen. [Mi17] Teilweise werden auch die Auswirkungen digitaler Medien oder der Automatisierung auf die Gesellschaft thematisiert. [Mi14]

4 Roboter

Folgend werden einige Roboterkonzepte vorgestellt, die für den Einsatz im Unterricht geeignet sein könnten. Darunter befinden sich kommerzielle sowie prototypische Projekte, wobei bei der Auswahl der Roboter eine hohe Diversität im Vordergrund stand.

Der **NIBObee** wird in den Programmiersprachen Java, Arduino, Assembler, C oder C++ programmiert. Er ist mit zwei Fühlern sowie drei Boden-/Liniensensoren ausgestattet. Angetrieben wird er durch zwei Motoren. Die Odometriemessung erfolgt mittels zweier Lichtschranken. [ni]
Für einen erweiterten Funktionsumfang ist es möglich folgende Module zu ergänzen: das *Blue-Modul* (eine ansteckbare Bluetooth-Einheit, zur Fernsteuerung über Android), ein schnellerer Mikrocontroller, ein Grafikdisplay, LEDs oder sogar einen Raspberry Pi. [ni]

Das **Hummingbird Kit** ist ein Roboterbausatz, bei dem die Roboter aus unterschiedlichen Komponenten zusammengestellt werden können. Das Gehäuse kann aus Pappe gestaltet werden. Außerdem stehen verschiedene Sensoren und Aktuatoren zur Verfügung. Dazu zählen Servomotoren, Getriebemotoren, Vibrationsmotoren, verschiedene LEDs sowie Sensorik für die Messung und Erkennung von Licht, Temperatur, Distanz und Lautstärke. [Kia] Es kann sowohl in den visuellen Sprachen *VisualProgrammer, Snap!, Scratch, BirdBlox* oder *Ardublock*, als auch in den Hochsprachen(*Python Java*) programmiert werden.[Kib]

Der **Arduino 2WD** ist ein einfach gehaltener Roboterbausatz. Er basiert auf einem *Arduino Uno Board*-Mikrocontroller. Zum Bausatz gehören zwei Antriebsräder, zwei Motoren und eine Gehäuseplatte. Arduino kann durch viele Komponenten erweitert werden. Der Roboter wird in der Programmiersprachenerweiterung Arduino für C oder C++ programmiert [Arc]. [di].

Bei dem **IOIO SHR** [Pe12] handelt es sich um ein privates Projekt, das nur als Bausatz verfügbar ist. Der Roboter selbst wird von einem Android-Smartphone gesteuert. Demzufolge

erfolgt die Programmierung über eine Android App in Java. Der Roboter wird durch Servomoten angetrieben, und mithilfe von diversen Sensoren, wie Taster, Liniensensoren und Compund Eye Sensor kann die Umwelt wahrgenommen werden.

Der ***Low Cost Robot* für das Erlernen der Java Programmierung** ist ein theoretischer Entwurf eines Roboters, über den momentan keine Informationen einer Kommerzialisierung oder Baupläne vorliegen. Der Antrieb und der Lenkmechanismus sind durch Antriebsservos realisiert. Der Roboter ist mit Ultraschallsensoren ausgestattet. Zur Kommunikation verfügt er über Nahfeld-Infrarotsensoren, ein Kontakband und ein Funkmodul. [SG09] Des Weiteren besitzt der Roboter einen Infrarotempfänger, um Signale von Sendern am Spielfeldrand zu empfangen und dadurch eine Positionsbestimmung zu ermöglichen. Die Programmierung des Roboters erfolgt in Java, zusätzlich stehen vorgefertigte Codeblöcke zur Verfügung.

Der **Lego Mindstorms EV3** besteht im Wesentlichen aus standardisierten Komponenten. Die modulare Bauweise erlaubt es verschiedene Robotermodelle zu realisieren [LEb].
Das Einstiegspaket enthält zusätzlich einen Mikrocontroller, einen Farbsensor, einen Berührungssensor, einen Infrarotsensor zur Abstandsmessung sowie Motoren [LEa]. Weitere Sensoren, wie u.a. ein Gyrosensor, ein Temperatursensor, ein Winkel- und Rotationssensor oder ein Barometersensor (z.T. von Drittherstellern) sind separat erhältlich [Geb]. Die Programmierung der Roboter kann über eine von Lego selbst entwickelte graphische Programmiersprache oder in Hochsprachen (Java) erfolgen [LEf].

5 Lehrerweiterbildung

Für einen sinvollen Einsatz von Robotern im Unterricht müssen Lehrkräfte einen gewissen Wissensstand mitbringen. Im Folgenden sollen Weiterbildungsmöglichkeiten auf ihrer Eignung analysiert werden.

Roberta Die Roberta Initiative hat sich zum Ziel gesetzt Lehrkräfte bei der Vermittlung von MINT-Fächern zu unterstützen. [IA] Besonders bezüglich des Themas Roboter im Unterricht bietet diese Plattform einen Anlaufpunkt. Es werden sehr viele Materialien [Inb] für Lehrkräfte zur Verfügung gestellt, mit denen Unterrichtsstunden gestaltet oder erweitert werden können. [Ina] Roberta bietet zudem Trainingseinheiten für Lehrkräfte bei zertifizierten Roberta Trainern an. Für die Programmierung wird dabei die eigens entwickelte, cloudbasierte Entwicklungsumgebung *Open Roberta Lab* verwendet. Darüberhinaus wird auch Lehrmaterial anderer Initiativen und Verlage bereitgestellt [Inb], mit welchem es unter anderem möglich ist die Roboter in Hochsprachen zu programmieren [IA15].

Mindstorms Lego stellt ein *Education center* [LEc] bereit, in dem Materialien für Kindergärten, Grundschulen oder weiterführende Schulen angeboten werden. Weiterhin werden Workshops für Einsteiger angeboten [LEe], in welchen Lernkonzepte und der Umgang mit den Robotern vermittelt werden. Zudem ist eine E-learning Plattform [LEd] verfügbar, welche eine Vielzahl an Lernvideos für Lehrkräfte bereitstellt.

Hummingbird Das in Abschnitt 4 erwähnte Roboterprojekt von Hummingbird ist stark auf den Einsatz im Unterricht fokusiert. Mithilfe von zertifizierten Trainern können Lehrkräfte geschult und weitergebildet werden. [Hua] Zudem gibt der Hersteller eine Empfehlung heraus, in welchen Schulformen das Produkt im Unterricht eingesetzt werden kann und welche Programmiersprache empfohlen wird. [Hub] Dabei kann der Roboter sowohl in visuellen- als auch in Hochsprachen (Python, Java) programmiert werden. Der Hersteller bietet zusätzlich Online-Kurse [Hud] und Tutorials [Huc] an.

Arduino Die Produkte des Herstellers Arduino werden in einer Vielzahl von Robotern und anderen Projekten verwendet. [Ard] Das Unternehmen Arduino stellt ein Portal mit Material für den Unterricht und für Weiterbildungen bereit. [Arb] Die Hauptkomponente dieses Lehrportals ist ein ausführlicher Onlinekurs, das sogenannte „Classroom 101“ [Ara] Programm. Es ist speziell auf den Einsatz in Schulklassen zugeschnitten.

6 Auswertung

Im Informatik-Unterricht der Oberstufe sind Roboter vor allem in den Themengebieten *Algorithmen und Datenstrukturen* und *Softwaretechnik* sinnvoll einzusetzen, da die Schüler je nach Können und Sensorik der Roboter Algorithmen entwerfen und implementieren können. Zum Beispiel wäre es hier möglich über verschiedene Ansätze der Breiten- oder Tiefensuche den Roboter unter Einsatz der Ultraschallsensoren den Weg durch ein Labyrinth finden zu lassen. Auch ließe sich die Thematik *Rechnernetze und Kommunikation* mit Hilfe von Robotern interaktiv umsetzen, wenn die Roboter mit Sensorik für die Kommunikation untereinander ausgerüstet sind.
In Tabelle 1 sind die in Kapitel 4.2 vorgestellten Roboter nach der Eignung für den Einsatz des jeweiligen Themas bewertet. Für den Themenbereich *Algorithmen und Datenstrukturen* schneidet ein Roboter in der Bewertung besser ab, wenn er möglichst viele verschiedene Sensoren besitzt, auf deren Grundlage Algorithmen entwickelt werden können. Für den Themenbereich *Softwaretechnik* wird ein Roboter aufgrund seiner Programmiersprachen bewertet. Außerdem spielt auch hier die Anzahl der Sensorik und die damit verbundene Komplexität der zu lösenden Aufgaben eine Rolle. Für die Thematik *Rechnernetze und Kommunikation* ist die Betrachtung und Bewertung der Sensorausstattung für die Kommunikation vorrangig. Neben den aus den Lehrplänen entnommenen Themengebieten, soll der Roboter auch aufgrund der Weiterbildungsmöglichkeiten für Lehrer bewertet werden, da nicht jeder Informatiklehrer umfangreiche Kenntnisse in der Robotik besitzt.

Nach diesen Bewertungsparametern geht der Lego Mindstorm EV3 als Sieger hervor. Durch seine große Anzahl an Sensoren inklusive derer für die Kommunikation und durch die hervorragenden Weiterbildungsmöglichkeiten für Lehrer hat er seinen Mitstreitern im Umfang Einiges voraus. Ähnlich sieht es mit dem Hummingbird Kit aus, welches sich durch ein umfangreiches Spektrum an Individualisierungsmöglichkeiten auszeichnet. Im Vergleich dazu verfügt der NIBObee über drei Sensoren. Obwohl er scheinbar keine Kommunikationsmöglichkeiten implementiert, kann der NIBObee an Schulen eingesetzt

	Algorithmen und Datenstrukturen	Softwaretechnik
NIBObee	Tastsensoren, Odometriesensoren und Liniensensor	Objektorientierte und leistungsfähige Programmiersprachen
Hummingbird Kit	Lichtsensor, Temperatursensor, Abstandssensor, Geräuschsensor und Drehgeber	Große Anzahl an unterschiedlichen Programmiersprachen
Arduino 2WD	Ultraschallsensoren und Raddrehzahlsensoren	Angelehnt an C++
IOIO SHR	Tastsensoren, Liniensensor, Compo- und Eye Sensor und Fototransistoren	Java
Java Low Cost Robot	Ultraschallsensoren, Nahfeld-Infrarotsensor & Kontaktband	Java
Lego Mindstorm EV3	Farbsensor, Infrarotsensor, Gyrosensor, Temperatursensor, Winkelsensor, Rotationssensor und Barometersensor	Visuelle Programmiersprache und Java

Tab. 1: Vergleich der Roboter

	Kommunikation und Rechnernetze	Weiterbildungsmöglichkeiten für Lehrer
NIBObee	keine	Tutorial vorhanden, kein Lehrermaterial
Hummingbird Kit	keine	Fortbildungen und Lehrmaterial
Arduino 2WD	Gute Erweiterbarkeit wegen der offenen Arduino Plattform, Kommunikation unter mehreren Robotern	Große Plattform für Lehrer von Arduino (Kurse + Materialien)
IOIO SHR	Sensoren des Smartphones	Wenig Tutorials, kein Lehrermaterial
Java Low Cost Robot	IR Empfänger und Funkmodul zur Kommunikation unter mehreren Robotern	Wenig Materialien, eine Beispiel Aufgabe vorhanden
Lego Mindstorm EV3	IR-Sensor dient auch dem Empfang von IR-Signalen; Funkmodul separat ehältlich [Gea]	Roberta

Tab. 2: Vergleich der Roboter 2

werden, bspw. wenn Schüler objektorientiert programmieren lernen sollen. Dem gegenüber steht der Arduino 2WD, der ebenfalls genug Sensorik bietet, um Schüler in der Mittelstufe in die Welt der Informatik einzuführen. Dies konnten die Autoren in einer Klasse der Junior-Ingenieur-Akademie im April 2018 beobachten. Als Aufgabe ist den Schülern in diesem Kurs gestellt den Roboter so zu programmieren, dass dieser autonom durch ein Labyrinth fährt. Einen ähnlichen spielerischen Ansatz wählt auch der Java-Roboter. Hiermit ist die Implementierung eines Fangspiels möglich, welches die Kommunikation unter den Robotern voraussetzt. Dies mit einer objektorientierten Programmiersprache wie Java erfüllt den Informatiklehrplan in einigen Themenschwerpunkten. Bei dem Roboter handelt es sich jedoch momentan lediglich um ein Konzept, welches noch nicht praktisch realisiert wurde. Der Roboter IOIO SHR ist durch die Programmierung und Steuerung über das Smartphone eine sehr interessante Idee. Durch die Programmierung als Android App mit Java wird die Robotik mit der Themenwelt der Smartphones vernetzt, mit welcher die Schüler zumeist bestens vertraut sind. Dies könnte das Interesse und die Motivation im Informatikunterricht weiter steigern. Die unausgereiften Lehrmaterialien erschweren jedoch die Verwendung im Unterricht. Des Weiteren setzt dieser Roboter voraus, dass jeder Schüler ein Android Smartphone besitzt und auch bereit ist, dieses im Unterricht einzusetzen, wobei eine mögliche Zerstörung des Smartphones nicht ausgeschlossen werden kann.
Die Betrachtung der verschiedenen Roboter zeigt insgesamt, dass Roboter sinnvoll im Informatikunterricht einsetzbar sind. Egal, ob der Lehrplan oder der Lehrer selbst den Schwerpunkt auf eine bestimmte Programmiersprache setzt oder ein bestimmtes Thema im Bereich der Algorithmen und der Softwaretechnik besonders ausführlich behandeln will, bietet der Einsatz von Robotern einen perfekten praktischen Ansatz. Des Weiteren bieten die oftmals umfangreichen Weiterbildungsmöglichkeiten für Lehrer beste Umstände, um den Informatikunterricht zu revolutionieren, ohne dass die Bundesländer die Lehrpläne anpassen müssten oder die Ausbildung der Informatiklehrer selbst angepasst werden muss.

7 Fazit und Ausblick

In dieser Arbeit konnte gezeigt werden, dass durchaus mehrere Robotermodelle existieren, die sich mit entsprechender Vorbereitung des Lehrpersonals nahtlos in den Schulunterricht integrieren lassen, ohne dass eine Anpassung der Curricula erforderlich ist.
Roboter als Lehr- und Lernobjekt eignen sich daher aus fachlicher Sicht gut als Teilmaßnahme, um dem Informatikunterricht in Deutschland aus der Krise zu helfen, da sie den Unterricht für die Schüler attraktiver und anwendungsorientierter machen, was in der Folge zu einer höheren Belegungsrate des Faches führen sollte. Aktuell gibt es jedoch nur wenig belastbare Forschung, welche die pädagogische Seite dieses Ansatzes näher betrachtet: Werden unter dem Einsatz von Robotern im Unterricht gleich gute oder gar bessere Lernergebnisse erzielt als im klassischen Informatikunterricht oder müssen ggf. besondere Anpassungen am Lehrkonzept vorgenommen werden, um einen solchen Unterricht praktikabel zu machen? Wie müssten diese aussehen? Auf diesem Gebiet wird dringend Grundlagenforschung benötigt.

Literatur

[17a] Hochschul-Bildungs-Report 2020 - HÖHERE CHANCEN DURCH HÖHERE BILDUNG?, Jahresbericht 2017/18 – Halbzeitbilanz 2010 bis 2015, 30. Juni 2017.

[17b] Fachkräfteengpassanalyse, Bundesagentur für Arbeit Statistik/Arbeitsmarktberichterstattung, Dezember 2017.

[17c] Das Schulwesen in Nordrhein-Westfalen aus quantitativer Sicht, Version 1, Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen, 21. März 2017.

[17d] Zahl der Studierenden steigt im Wintersemester 2017/2018 erneut an, https://www.destatis.de/DE/PresseService/Presse/Pressemitteilungen/2017/11/PD17_427_213.html, Statistisches Bundesamt, 28. Nov. 2017, Stand: 29. 04. 2018.

[17e] Schule digital - Der Länderindikator 2017, Digitale Medien in den MINT-Fächer, Deutsche Telekom Stiftung, Nov. 2017.

[Ara] Arduino: Classroom 101, https://store.arduino.cc/arduino-ctc-101-program, Stand: 27. 04. 2018.

[Arb] Arduino: Education, http://de.wikipedia.org/w/index.php?title=Plagiat&oldid=43117308, Stand: 27. 04. 2018.

[Arc] Arduino: Frequently Asked Questions, https://www.arduino.cc/en/Main/FAQ, Stand: 25. 04. 2018.

[Ard] Arduino: Project Hub, https://create.arduino.cc/projecthub, Stand: 27. 04. 2018.

[Be17a] Berg, A.: Der Arbeitsmarkt für IT - Fachkräfte, Berlin, 7. Nov. 2017.

[Be17b] Bergner, N.: Digitale Bildung in der Schule – die Lehrkräfte sind der Schlüssel. Material- und Fortbildungsangebote zum Thema digitales Lernen, http://www.medienpaed.com/article/view/474, 2017.

[ÇDY17] Çankaya, S.; Durak, G.; Yünkül, E.: Education on Programming with Robots: Examining Students' Experiences and Views. Turkish Online Journal of Qualitative Inquiry/, Oktober 2017.

[Ch15] Chetty, J.: The notion of Lego© Mindstorms as a powerful pedagogical tool: Scaffolding learners through computational thinking and computer programming./, 2015.

[De17] acatech – Deutsche Akademie der Technikwissenschaften KörberStiftung, D. g.: MINT Nachwuchsbarometer 2017, https://www.koerber-stiftung.de/fileadmin/user_upload/koerber-stiftung/redaktion/mint_nachwuchsbarometer/pdf/2017/MINT-Nachwuchsbarometer-Langfassung.pdf, 2017.

[di] diyelectronics: Build guide to basic Arduino 2WD robot car, https://www.diyelectronics.co.za/wiki/robotcarkit/, Stand: 25. 04. 2018.

[Dü17] Dübener, S. U.; Morgner, A. A.; Haupt, H. F.; Volk, M. H.; Fischer, C. C.; Langner, S. K.; Jacker, A.: Gegenüberstellung von kostengünstigen Robotern als Lernobjekte für Schulen. In (Eibl, M.; Gaedke, M., Hrsg.): INFORMATIK 2017. Gesellschaft für Informatik, Bonn, S. 2461–2472, 2017.

[Fr09] Freie und Hansestadt Hamburg - Behörde für Schule und Berufsbildung: Bildungsplan gymnasiale Oberstufe - Informatik, 2009.

[Gea] Generationrobots: Drahtloses Kommunikationsmodul NXTBee Pro, https://www.generationrobots.com/de/181-lego-mindstorms-sensorenDrahtlosesKommunikationsmodulNXTBeePro, Stand: 28. 04. 2018.

[Geb] Generationrobots: Lego Mindstorms Sensoren, https://www.generationrobots.com/de/181-lego-mindstorms-sensoren, Stand: 28. 04. 2018.

[He16] Hessisches Kultusministerium: Kerncurriculum gymnasiale Oberstufe - Informatik, 2016.

[Hua] Humminbird: Professional Development Opportunities, http://www.hummingbirdkit.com/teaching/training, Stand: 26. 04. 2018.

[Hub] Humminbird: Recommended Software, http://www.hummingbirdkit.com/learning/software, Stand: 26. 04. 2018.

[Huc] Humminbird: Tutorials, http://www.hummingbirdkit.com/learning/tutorials, Stand: 26. 04. 2018.

[Hud] Humminbird: Virtual Training Workshop, http://www.hummingbirdkit.com/teaching/training/virtual-workshop, Stand: 26. 04. 2018.

[IA] IAIS, F.: MINT-Förderung und Bildung, https://www.iais.fraunhofer.de/de/geschaeftsfelder/intelligente-medien-und-lernsysteme/uebersicht/mint-foerderung-und-bildung.html, Stand: 26. 04. 2018.

[IA15] IAIS, F.: EV3-Programmieren mit Java, https://www.roberta-home.de/fileadmin/user_upload/Roberta-EV3programmierenJava_small.pdf, 2015.

[Ina] Initiative, R.: Lerneinheiten und Experimente, https://www.roberta-home.de/lehrkraefte/lerneinheitenexperimente/, Stand: 26. 04. 2018.

[Inb] Initiative, R.: Roberta Materialien, https://www.roberta-home.de/lehrkraefte/roberta-materialien/, Stand: 26. 04. 2018.

[JTA15] Janzen, I.; Thomas, M.; Angélica, Y.: Wahlverhalten zum Schulfach Informatik in der SI - eine Studie im Regierungsbezirk Münster, 2015.

[Kia] Kit, H. R.: Hummingbird Duo Datasheet, https://www.hummingbirdkit.com/learning/duo-datasheet, Stand: 25. 04. 2018.

[Kib] Kit, H. R.: Software, https://www.hummingbirdkit.com/learning/software, Stand: 25. 04. 2018.

[La06] Land Brandenburg: Kerncurriculum für die Qualifikationsphase der gymnasialen Oberstufe - Informatik, 2006.

[LEa] LEGO: 31313 MINDSTORMS EV3, https://www.lego.com/de-de/mindstorms/products/mindstorms-ev3-31313, Stand: 28. 04. 2018.

[LEb] LEGO: Baue einen Roboter, https://www.lego.com/de-de/mindstorms/build-a-robot, Stand: 28. 04. 2018.

[LEc] LEGO: LEGO Education, https://education.lego.com/de-de, Stand: 26. 04. 2018.

[LEd] LEGO: LEGO Education Academy, https://elearning.legoeducation.com/, Stand: 26. 04. 2018.

[LEe] LEGO: Lehrer-Fortbildungen und Kennenlern-Workshops, https://education.lego.com/de-de/lehrerfortbildungen, Stand: 26. 04. 2018.

[LEf] LEGO: LERNE PROGRAMMIEREN – DAS GEHT GANZ LEICHT, https://www.lego.com/de-de/mindstorms/learn-to-program, Stand: 28. 04. 2018.

[Mi02] Ministerium für Bildung, Wissenschaft, Forschung und Kultur des Landes Schleswig-Holstein: Lehrplan für Sekundarstufe II - Informatik, 2002.

[Mi11] Ministerium für Bildung, Jugend und Sport des Landes Brandenburg: Vorläufiger Rahmenlehrplan für den Unterricht in der gymnasialen Oberstufe im Land Brandenburg - Informatik, 2011.

[Mi14] Ministerium für Schule und Weiterbildung des Landes Nordrhein-Westfalen: Kernlehrplan für die Sekundarstufe II Gymnasium/Gesamtschule in Nordrhein-Westfalen - Informatik, 2014.

[Mi17] Ministerium für Bildung Sachsen Anhalt: Fachlehrplan Gymnasium - Informatik, 2017.

[ni] nibo-roboter: NIBObee, http://www.nibo-roboter.de/wiki/NIBObee, Stand: 25. 04. 2018.

[Pe12] Peter: IOIO-SHR – Java programmierbarer Roboter, https://www.robotfreak.de/blog/ioio-shr-java-programmierbarer-roboter/, 2. Jan. 2012, Stand: 25. 04. 2018.

[Rh11] Rheinland-Pfalz Ministerium für Bildung, Wissenschaft, Jugend und Kultur: Lehrplan Informatik - Mainzer Studienstufe, 2011.

[Sa06] Saarland Ministerium für Bildung, Kultur und Wissenschaft: Achtjähriges Gymnasium - Informatik, 2006.

[Sä11] Sächsisches Staatsministerium für Kultus und Sport: Lehrplan Gymnasium - Informatik, 2011.

[Se09] Senatorin für Bildung und Wissenschaft: Informatik Bildungsplan für die Gymnasiale Oberstufe, 2009.

[Se10] Senatsverwaltung für Bildung, Jugend und Sport Berlin: Rahmenlehrplan für die gymnasiale Oberstufe - Informatik, 2010.

[SG09] Samuelsen, D. A. H.; Graven, O. H.: Low Cost Robots as Target System for Students Training Using Java. International Journal of Online Engineering/, 2009.

[SS15] Schauer, C.; Schauer, H.: Schulische IT- und Medienbildung: Ergebnisse einer empirischen Studie an einem rheinland-pfälzischen Gymnasium, ger, http://hdl.handle.net/10419/116780, ICB-Research Report, Essen, 2015.

[Th12] Thüringer Ministerium für Bildung, Wissenschaft und Kultur: Lehrplan für den Erwerb der allgemeinen Hochschulreife - Informatik, 2012.

Von der (Nicht-)Intelligenz der Algorithmen

Die Frage nach der Existenz von intelligenten Algorithmen mit Hilfe des Intelligenzverständnisses von Gilbert Ryle (1969)

Christopher Klamm[1]

Abstract: Superintelligent oder gottesgleich, in der Debatte um künstliche Intelligenz werden Algorithmen verstärkt zu transzendentalen Wesen erhoben. Die Debatten setzten dabei voraus, dass es künstliche Intelligenz überhaupt gibt. Diese Arbeit betrachtet, ob Algorithmen wirklich intelligent sind, oder, ob die Debatte verfrüht eine Intelligenz annimmt, die es (noch) nicht gibt. Anhand des Intelligenzverständnisses nach Gilbert Ryle werden algorithmische Ausprägungen künstlicher Intelligenz prüfend betrachtet. Sein Verständnis agiert als Prüfstein und dient zur Betrachtung von (musterbasierten) Algorithmen und neuronalen Netzen. Es zeigt sich dabei, dass aktuelle Algorithmen eine zunehmende Komplexitätsstufe besitzen sowie eine Vielzahl von intelligenten Merkmalen in sich tragen, jedoch letztlich (noch) am Intelligenzverständnis nach Ryle scheitern, da sie die Lerngrenzen ihrer Verfasstheit nicht überwinden.

Keywords: Intelligenz, künstliche Intelligenz, Algorithmen, Gilbert Ryle

1 Einleitung

„Alles wird intelligent“, so das Postulat der Forscher im digitalen Manifest 2015 [He15]. Algorithmen beeinflussen unser Leben. Googles DeepMind-Algorithmen können autonom Spiele spielen (z.B. das Brettspiel Go) [Be16], Algorithmen können Schrift, Sprache und Muster fast so gut erkennen wie Menschen und viele Aufgaben sogar besser lösen. Es wird immer häufiger von „Superintelligenz“ gesprochen [He15]. Diese künstliche Intelligenz ist allgegenwärtig, mit einem immensen wirtschaftlichen Wachstum [Ca17, p. 138]. Algorithmen, die Grundlage dieser Intelligenz, sind durch ihre Vielschichtigkeit ein Quell der Angst und Begeisterung. Es ist davon die Rede, dass sie „gottesgleiche[s] Wissen“ [Wo15] besäßen und „erschreckend intelligent“ [Gr15] seien. Ebenso aber auch vom Problem, dass diese über unser Verständnis hinaus gehen – „wir wissen nicht, wie sie’s tun“ [An17] – und „bald mehr [können] als wir ihnen beibringen“ [Bu17]. Technologievisionäre wie Elon Musk von Tesla Motors, Bill Gates von Microsoft und Apple-Mitbegründer Steve Wozniak warnen. Dirk Helbing [He16] fragt in diesem Kontext „Maschinelle Intelligenz – Fluch oder Segen?“ – doch gibt es bereits eine *maschinelle Intelligenz!* oder müssen wir noch fragen: *maschinelle Intelligenz?*

[1] TU Darmstadt, Hochschulstraße 10, 64289 Darmstadt, christopher.klamm@stud.tu-darmstadt.de

Im Grundgedanken der künstlichen Intelligenz, deren führender Begründer u.a. der Informatiker John McCarthy war, ist es „the science and engineering of making intelligent machines“ [Mc07]. Eine Maschine besitzt eine Aufgabe, welche diese bearbeitet. Sie soll uns z.B. in Alltagssituationen helfen und das Licht zur richtigen Zeit anschalten oder Berufstätige bei ihrer Arbeit unterstützen. Aber ab wann genau tut sie dies intelligent? Was heißt es, etwas intelligent zu tun? Eine Maschine ist die Summe ihrer Fähigkeiten. Um davon zu sprechen, dass eine Maschine intelligent ist, müsste jede dieser Handlungen auf eine intelligente Weise ausgeführt werden. Finden wir nur ein Indiz der *Nicht-Intelligenz* innerhalb einer Maschine, dann können wir Karl Popper [Po00] und den Falsifikationisten folgend bereits davon ausgehen, dass es keine intelligente Maschine ist. Im Digitalen besteht eine künstliche Maschine bzw. künstliche Intelligenz aus einer Vielzahl von Algorithmen, die ineinander verflochten sind[2]. Wenn wir wissen wollen, ob eine digitale Maschine intelligent ist, dann müssen wir uns anschauen, ob ihre zugrundeliegenden Algorithmen intelligent sind bzw. handeln[3]. Wir müssen uns dazu also die Frage stellen: *Was ist ein intelligenter Algorithmus und gibt es bereits heute intelligente Algorithmen?*

Ich werde argumentieren, dass es nicht den *einen* Algorithmus gibt und somit die Form der Intelligenz unterschiedlich ausgeprägt ist. Ebenfalls soll argumentativ dargelegt werden, dass eine fortwährende Erweiterung intelligenter Elemente in Algorithmen vorzufinden ist, es aber aktuell noch keine intelligenten Algorithmen gibt, da der Mensch selbst noch als *intelligenter Konstrukteur* eingreift. Um dies aufzuzeigen, sollen folgend zwei Aspekte erörtert werden. Als erstes müssen wir ein Verständnis darüber gewinnen, was es bedeutet, dass eine *Handlung intelligent* ist. Wir benötigen ein *Kriterium*, dass wir prüfend anlegen können. Dieses soll anhand des Verständnisses einer intelligenten Handlung nach Gilbert Ryle [Ry69] erarbeitet werden[4]. Anschließend können wir damit eine erste dichotome Unterscheidung (intelligent/ nicht intelligent) vornehmen. Neben dem Maßstab muss der Untersuchungsgegenstand selbst beleuchtet werden, der *Algorithmus*. Dabei ist zu erarbeiten, ob es den einen Algorithmus gibt oder ob dies eine vielschichtige Begrifflichkeit ist, die in ihren Facetten gesondert zu behandeln ist. Mit Hilfe dieses Wissens können wir unser Kriterium an das Verständnis eines Algorithmus anlegen und aufklären, ob es den intelligenten Algorithmus überhaupt gibt. Damit ergründen wir, ob die aktuelle Debatte begrifflich noch zu weit greift.

---

[2] Dies folgt dem Verständnis von McCarthy (2007), welcher die Maschine mit einem Computerprogramm bzw. einer Summe von Algorithmen gleichsetzt [Mc07, p. 2].

[3] Wir gehen hier dem Falsifikationismus folgend davon aus, dass die intelligente Maschine – auf der Basis ihrer zugrundeliegenden Algorithmen – auf ihre Intelligenz hin überprüft werden kann. Dies schließt eine intelligente Maschine aus nur nicht-intelligenten Algorithmen per Methode aus.

[4] Ryle ist besonders gewinnbringend, da er Intelligenz mit der Art und Weise einer Handlung verknüpft und damit die Betrachtung eines handlungsorientierten Algorithmus ermöglicht. Dies ist wichtig, da ein Algorithmus in der Arbeit nicht als eine starre, atomare Struktur verstanden werden soll. Er ist zutiefst handlungsorientiert – nicht die atomare Einheit ist intelligent oder nicht-intelligent, sondern seine Art und Weise Probleme zu lösen.

2 Eine *intelligente Handlung* nach Gilbert Ryle

Gilbert Ryle macht die Frage nach einer intelligenten Handlung zu seinem Untersuchungsgegenstand im Werk „Der Begriff des Geistes“ (1969). Der gedankliche Beginn liegt bei Ryle im *intellektuellen Missverständnis*. Dieses gehe davon aus, dass eine Handlung nur dann intelligent ist, wenn ihr ein (intellektuelles) Abwägen von handlungsleitenden Sätzen vorangeht (Ryle zeigt u.a. das Problem des unendlichen Regresses, siehe dazu [Ry69, p. 35]). Gegen dieses Missverständnis schreibt er an. Dies ist für ihn offenkundig und er zeigt dies an zahlreichen Beispielen, die für sich intelligente Handlungen sind, denen aber dieses vorherige Abwägen nicht innewohnt. Eines seiner Beispiele ist – neben dem Schachspielen – das Argumentieren. Er hält dazu fest: „Regeln für richtiges Argumentieren wurden zuerst von Aristoteles ausgearbeitet, aber Leute konnten Fehlschlüsse vermeiden und aufdecken, lange bevor sie seine Lehren darüber gelernt hatten [...]“ [Ry69, p. 33]. Dies gehe somit, so Ryle, auch ohne ein gebetsmühlenartiges Vorhersagen der Spielregel vonstatten. Ryles Ziel ist es zu zeigen, dass die Intelligenz, als eine Fähigkeit zum praktischen Handeln, und der Intellekt, als Fähigkeit zum Theoretisieren („Ziel dieser Tätigkeiten ist die Erkenntnis von wahren Sätzen oder Tatsachen“ [Ry69, p. 28]), entgegengesetzt dem intellektuellen Missverständnis, zu verstehen sind. Das bedeutet, dass die Fähigkeit zum Theoretisieren selbst nur eine praktische Fähigkeit ist – „Das Erwägen von Sätzen ist selbst eine Tätigkeit, die mehr oder weniger intelligent [...] ausgeführt werden kann“ [Ry69, p. 34]. In dieser gehe es auch darum die Regeln klug anzuwenden, auch ohne inneren Monolog. Ryle unterscheidet dabei zwischen Wissen und Können, um zu verdeutlichen, dass jedem Wissen ein Können vorausgehen muss – „Erfolgreiche Praxis geht ihrer eigenen Theorie voraus [...]“ [Ry69, p. 33].

Intelligent sei jemand, der fähig ist, gewisse Dinge zu tun [Ry69, p. 30]. „[E]r [der Mensch] führt seine Tätigkeiten wirksam aus [...]. Es heißt: ein Ding auf eine bestimmte Weise tun, oder es in einem gewissen Stil oder nach einem gewissen Verfahren zu tun [...]“ [Ry69, p. 58f.]. Seine Handlung hat eine gewisse Qualität bzw. sie erreicht ein „gewisses Niveau“ [Ry69, p. 31]. Solch eine Handlung basiert nicht auf Gewohnheit. Er hält dazu fest: „Wenn wir von jemandem sagen, er tue etwas aus reiner oder blinder Gewohnheit, dann meinen wir, er tue es automatisch und ohne dabei auf das, was er tut, achten zu müssen. [...] Es gehört zum Wesen der bloß gewohnheitsmäßigen Handlung, daß Einzelverrichtungen der Abklatsch ihrer Vorgänger sind. Es gehört zum Wesen intelligenter Handlung, daß Einzelverrichtungen durch ihre Vorgänger beeinflußt werden“ [Ry69, p. 50].

Der Schüler „lernt, wie man etwas denkend macht, so daß jede einzelne Handlung selbst eine neue Unterweisung bedeutet“ [Ry69, p. 51]. Ausbildung ist es, welche die Intelligenz schafft, eine bloße Abrichtung verzichtet auf diese (ebd.). Der Handelnde muss in der Lage sein, „[...] Neuerungen einzuführen, und wo er das tut, handelt er nicht aus Gewohnheit“ [Ry69, p. 57]. Es ist wichtig, dass „[...] er fähig ist, in seinem Vorgehen Fehler zu entdecken und auszumerzen, Erfolge zu wiederholen und zu vergrößern, aus den Beispielen anderer zu lernen und so weiter“ [Ry69, p. 31]. Dies zeigt bereits, dass wir einer einzelnen „Handlung allein [...] nicht mit Sicherheit ansehen, ob die Ausübung eine Fertigkeit ist“

[Ry69, p. 54], z.B. der gezielte Schuss ins Schwarz beim Schießen (ebd.). Jemand der etwas Intelligentes tut, ist verschiedenen von dem, der zufällig handelt. Es ist die Fähigkeit, Handlungen auf diese Art durchzuführen – jene sind intelligent, die „für ihre Handlungen verantwortlich sind. Intelligent sein heißt nicht bloß gewissen Kriterien zu genügen, sondern sie anzuwenden“ [Ry69, p. 31]. Damit besitzt er eine „Disposition, [...] solche Handlungen richtig, erfolgreich usw. auszuführen“ [Ke75, p. 164]. Wie der Bergsteiger im Dunkeln, der „mit einem gewissen Maß von Geschicklichkeit und Verstand“ vorgeht [Ry69, p. 50]. Oder der intelligente Kraftfahrer: „[d]en durchgegangenen Esel hat er nicht vorausgesehen, aber ist auf ihn gefaßt“ [Ry69, p. 58]. Der Handelnde hat die Gedanken bei der Sache. Eine solche Fähigkeit zu besitzen bedeutet, eine mehrspurige Disposition zu richtigem Verhalten zu besitzen [Ke75, p. 164]. „Können ist also eine Disposition, aber nicht eine eingleisige Disposition wie ein Reflex oder eine Gewohnheit. Es wird in der Befolgung von Regeln oder Richtlinien oder der Anwendung von Kriterien ausgeübt [...]“ [Ry69, p. 56]. Die „höherstehenden menschlichen Dispositionen [...] [sind] [...] im Allgemeinen nicht eingleisig“ [Ry69, p. 53], nicht wie beispielweise das Weinglas die Disposition zerbrechlich besitzt [Ry69, p. 51]. Hinter aller Intelligenz steht implizit die Logik als Grundprinzip, denn, wenn er intelligent handelt „[...] befolgt er die Regeln der Logik, [wahrscheinlich] ohne an sie zu denken“ [Ry69, p. 58]. Intelligente Handlung erfährt eine Kennzeichnung. Die „Beschreibung dieses modus operandi muß mit Hilfe solcher halb dispositionellen, halb episodischen Beiwörter wie ‚wachsam‘, ‚sorgfältig‘, ‚kritisch‘, ‚genial‘, ‚logisch‘ gekennzeichnet werden.“ [Ry69, p. 58f.]

Um den Untersuchungsgegenstand *Algorithmus* mit einem Maßstab zu prüfen, sollen die Darstellungen nach Ryle in einer Art *Intelligenz-Prüfstein* zusammengefasst werden. Dieser besteht aus denen von Ryle ausgeführten Elementen. Als erstes muss eine Handlung, um intelligent zu sein, nicht aus Gewohnheit entstehen, kein bloßer Abklatsch sein, Neuerungen müssen somit möglich sein. Ebenfalls ist es wichtig, dass eine intelligente Handlung absichtsvoll und nicht aus reinem Zufall passiert. Das Ablaufen einer Handlung muss dabei stets logischen Gesetzmäßigkeiten folgen. Abschließend ist es wichtig, dass nur intelligent gehandelt werden kann, wenn auch aus den Fehlern gelernt wird und das ist damit komplementär zum bloßen Abrichten. Eine solche Handlung erhält nach Ryle die Beiwörter *sorgfältig, kritisch, genial* oder *logisch*. Diese angeführten Kriterien sind nicht disjunkt und überlagern sich teilweise in ihrer Bedeutung und sagen somit besonders etwas in ihrer Gesamtheit über eine Handlung aus.

Mit Ryle haben wir etwas gewonnen, mit dessen Hilfe wir uns ein Bild davon machen können, was es bedeuten kann, wenn von Intelligenz bzw. einer intelligenten Handlung gesprochen wird. Dieses Verständnis soll uns begleiten, wenn wir uns dem Algorithmus nähern und der Frage, ob ein Algorithmus wirklich intelligent ist. Doch ohne einen Begriff davon zu besitzen, was ein Algorithmus ist, haben wir nichts, woran wir diesen Maßstab anlegen können. Aus diesem Grund möchte ich den Begriff folgend näher betrachten.

3 Der Algorithmus als intelligenter Problemlöser

Betrachten wir Algorithmen, so wird die intuitive Vorstellung sich schnell im technischen Raum der Computer ansiedeln. Bevor wir zu einem technisierten Verständnis von Algorithmen vordringen, können wir Algorithmen viel gegenständlicher als eine Art *Handlungsvorschrift* begreifen – Pomberger/ Dobler [2008, p. 33] bezeichnen einen Algorithmus auch als „schrittweises Problemlösungsverfahren". Dieser besteht aus definiert endlich vielen Schritten und ist deterministisch. Bei der dadurch erzeugten Problemlösung wird eine bestimmte Eingabe in eine Ausgabe überführt. Ein sehr zentrales Verständnis eines Algorithmus ist damit seine *Funktion*, welche in der Problemlösung besteht bzw. eine Eingabe in eine Ausgabe zu überführen[5]. So verstanden ist ein Algorithmus eine Black-Box zwischen Eingabe und Ausgabe. Wichtig dabei ist, dass eine intendierte Ausgabe erzeugt werden soll, nämlich die Lösung eines Problems.

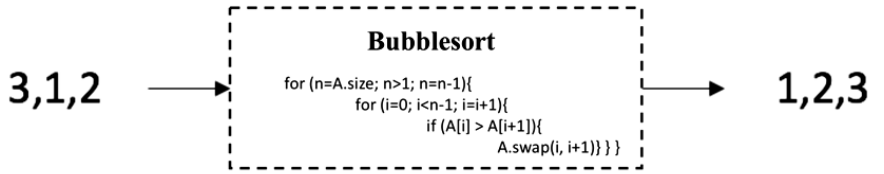


Abb. 1: Algorithmus "Bubblesort" (eigene Darstellung)

Wenn wir anschließend in eine technische Umsetzung eines Algorithmus hineinschauen, dann können wir dazu beispielweise einen Algorithmus aus einem Bereich der grundlegendsten Algorithmen wählen, wie dem Bereich der Sortieralgorithmen. Der Sortieralgorithmus *Bubblesort* markiert den Beginn derartiger Algorithmen (Abb. 1). Für diesen Algorithmus möchte ich zeigen, welche Funktion er ausübt und welche Ein- und Ausgabe dafür benötigt bzw. generiert werden. Als Intention steht, eine Liste von Zahlen zu sortieren z.B. 3, 1, 2. Damit wäre die Eingabe eine Liste von Zahlen, welche in eine Liste von Zahlen als Ausgabe überführt werden soll. In diesem Fall wäre die Eingabe 3, 1, 2 und die Ausgabe 1, 2, 3. Diese basale Operation verdeutlicht den Problemlösecharakter eines Algorithmus. Das Problem ist eine potenziell unsortierte Reihenfolge von Zahlen und die dazugehörige geordnete Lösung.

Ist diese Fähigkeit bereits eine intelligent ausgeführte Handlung, im Sinne Ryles? Um dies zu überprüfen, greifen wir auf den erarbeiten Prüfstein zurück. Der einfache Problemlöser Algorithmus ist vor allem eines: *nicht dynamisch*. Grund dafür ist, dass er durch einen menschlichen Konstrukteur einprogrammiert ist. Auf einem „neuen Berg" findet er sich nicht zurecht. Er ist vorgedacht, mit einem festen Plan, der unveränderlich ist. Er ist wie

[5] Dieses Verständnis verengt für die handhabbare Betrachtung die Komplexität eines Algorithmus, welcher auch über seine Eigenschaften: Korrektheit, Effizienz, dynamische Endlichkeit, Vollständigkeit, Eindeutigkeit, statische Endlichkeit oder Ausführbarkeit beschrieben werden könnte (siehe dazu z.B. [PD08, p. 33ff.]).

eine definierte Fahrroute, von der auch bei zu erwartender längerer Fahrzeit oder bei Baustellen nicht abgewichen werden würde. Jeder neue Sortierdurchlauf wäre somit lediglich ein Abklatsch. Während seiner Ausführung nimmt *Bubblesort* keine gelernte Veränderung seiner strukturellen Gestalt vor. Der Algorithmus ist jedoch nicht zufällig. Die definierte Reihenfolge schafft Verlässlichkeit. Bekommt der Algorithmus unsortierte Zahlen als Eingabe, dann ist die Ausgabe sortierte Zahlen. Ebenfalls folgt er selbst einer Logik. Diese ist für Zahlen auch auf einer mathematischen Logik begründet. Betrachten wir seine Fähigkeit aus Fehlern zu lernen[6], Neuerungen einzubauen, sich dynamisch zu verändern, dann ist dies für einen derartigen Algorithmus nicht gegeben. Er selbst würde die Fehler nicht bemerken, es wären für ihn keine. Wird dem Algorithmus eine Fähigkeit des Sortierens zugeschrieben, dann würden wir spätestens bei der Sortierung von Buchstaben feststellen, dass er diese nur eingeschränkt für Zahlen und nicht für Buchstaben besitzt. Einem derartigen Algorithmus mag man das Beiwort sorgfältig und in den meisten Fällen auch logisch anfügen, genial oder gar kritisch ist er nicht. Der einfache Algorithmus ist ein unveränderliches strukturelles Gebilde einer einprogrammierten Lösungsvorschrift, bei dem nach Ryle nicht von einem *intelligenten Algorithmus* gesprochen werden kann.

Damit könnten wir die Betrachtung von intelligenten Algorithmen bereits beenden. Doch damit würden wir der Vielfältigkeit von Algorithmen nicht gerecht werden. Das antiquierte Verständnis eines Algorithmus, als starre Lösungsvorschrift, kann heute nicht mehr aufrechterhalten werden. Während es diese elementare Form des Algorithmus nach wie vor gibt und selbst bei allen weiteren Algorithmen als Hintergrundrauschen existent ist, hat die Ausdifferenzierung und Erweiterung der Komplexität von Algorithmen ganz neue Formen hervorgebracht, welche wir nicht unbeachtet zurücklassen können.

4 Die Existenz intelligenter Algorithmen

In der heutigen Entwicklung spielen besonders Algorithmen eine wichtige Rolle, die musterbasiert Probleme lösen. Diese Form ist im ersten Schritt in der Lage, anhand von vorprogrammierten Mustern bzw. Merkmalen beliebige Daten anhand dieser zu betrachten. Und in einem zweiten Schritt anhand dieser Muster – auf Basis von Wahrscheinlichkeiten – Aussagen zu treffen. Das Problemlösen ist weiter die Aufgabe eines Algorithmus, was sich dabei ändert, ist seine Art, diese zu bewältigen. Während einfache traditionelle Problemlöser, wie im Vorangegangenen dargestellt, einen vordefinierten/ vorprogrammierten Pfad der Aufgabenbewältigung haben, sind die musterbasierten Algorithmen mit einer neuen Art von Beliebigkeit und Anpassungsfähigkeit ausgestattet. An einem Beispiel lässt sich diese Neuartigkeit verdeutlichen. Ein beliebtes und immer noch sehr aktuelles Problem ist die Bestimmung von Stimmungslagen, die mit einem Text ausgedrückt werden[7]. Nehmen wir das Satzbeispiel: *„Intelligente Algorithmen sind eine große Gefahr für unsere*

[6] Lernen meint hier, dass er in der Lage ist, seine Handlung nach Erfahrung/ Feedback anzupassen. Ein Fehler ist dabei ein negatives Feedback – hervorgerufen durch eine Handlung.

[7] siehe dazu z.B. *Sentiment Analyse* für kurze Texte [KWM11]

Menschheit". Dieser Satz bringt eine *negative* Stimmung gegenüber dem Thema Algorithmen zum Ausdruck. Die Aufgabe eines Algorithmus wäre es zu erkennen, dass dieser Satz negativ gemeint ist – auch ohne, dass wir dem Algorithmus genau sagen, dass dies der Fall ist. Seine allgemeine Aufgabe ist somit, für einen beliebigen Satz vorherzusagen, ob dieser positiv oder negativ gemeint ist. Um dies zu ermöglichen, versucht der Algorithmus Muster innerhalb von positiven und von negativen Sätzen wiederzufinden. Diese Muster werden durch den Menschen als intelligenten Konstrukteur einprogrammiert. Beispielsweise werden dafür sogenannte n-Gramme verwendet. Ein n-Gramm ist in einem Satz eine zusammenhängende Anzahl von z.B. Wörtern. Das einfachste n-Gramm ist das Unigramm. Das bedeutet, dass sich der Algorithmus immer genau ein Wort anschaut. Ein dadurch zu erkennendes Muster wäre, dass in einem negativen Satz z.B. das Wort *Gefahr* häufiger vorkommen könnte. Anhand von sehr vielen Beispielen würde der Algorithmus immer wieder schauen, ob das Wort *Gefahr* ein verlässlicher Indikator dafür ist, dass der Satz negativ ist. Der Algorithmus basiert dabei auf statistischen Modellen, wie dem populären *Logistic Regression Model* [Da50]. Dieses Model versucht mathematisch die bekannten Daten auf der Basis der Merkmale, wie z.B. der verwendeten Wörter, zu beschrieben. Weitere Merkmale, neben den einzelnen Wörtern, könnten Wortlänge oder Wortkombinationen, wie z.B. *große Gefahr,* sein[8]. Anhand einer Vielzahl von definierten Merkmalen lernt das Model einen unbekannten Satz richtig einzuordnen, indem es eine Wahrscheinlichkeit angibt, mit welcher ein Satz positiv/ negativ ist (siehe Abb. 2).

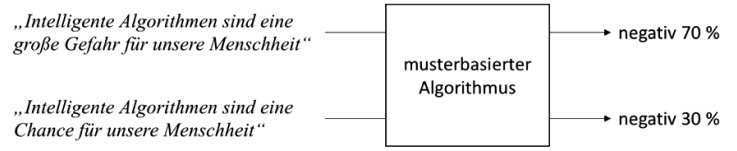


Abb. 2: "Sentiment Analyse" mit Hilfe eines musterbasierten Algorithmus (eigene Darstellung)

Ist diese neue Fähigkeit der über Muster definierten Algorithmen bereits ein Intelligenzindikator? Um dies zu prüfen, legen wir wieder den erarbeiten Prüfstein nach Ryle an. Musterbasierte Algorithmen agieren nicht zufällig. Die Entscheidung lässt sich logisch anhand der statistischen Betrachtung von Merkmalen über viele Beispiele hinweg ableiten. Schwieriger wird es bei der Frage der Gewohnheit. Die Muster sind durch den Entwickler vordefiniert und werden nicht gelernt. Der Algorithmus entscheidet auf der Basis der Muster. Würde er merken, dass er sehr schlecht im Vorhersagen ist, dann könnte er das nicht selbst ändern. Er lernt in seinem ihm gegebenen Lernhorizont, aber kann sich strukturell nicht verändern. Er ist auch, wie ein traditioneller Algorithmus, strukturell statisch angelegt. Dennoch agiert er nicht einfach aus Gewohnheit. Seine Entscheidung basiert auf einer Erfahrung. Sein Erfahrungshorizont ist zwar begrenzt, aber in diesem sammelt er Erfahrungen und kann diese auch in einer neuen Situation, die ihm unbekannt ist,

[8] In diesem Forschungsbereich ist besonders das Finden von geeigneten Mustern die wissenschaftliche Leistung (siehe dazu z.B. [So13]).

anwenden. Er geht jedoch mit seiner Beschaffenheit weit über die traditionellen Algorithmen hinaus. Er besitzt die Fähigkeit, kontextübergreifend für Unbekanntes Entscheidungen zu treffen. Von einer klassischen, vollständig vorgedachten Ablaufstruktur ist er somit weit entfernt. Dennoch wird auch hier deutlich, dass er nach unserem erarbeiteten Verständnis nicht intelligent handelt, da er nicht lernen kann, um Neuerungen gerecht zu werden. Er kennt zwar seine „Berge“ gut, hat aber nicht die Fähigkeit, sich auf strukturell anderes Gelände einzustellen.

Auch die Betrachtung von musterbasierten Algorithmen hat gezeigt, dass Algorithmen nicht intelligent sind und unsere These ein weiteres Mal widerlegt. Dennoch ist der Bereich der Algorithmen noch nicht ausschöpfend betrachtet worden. In den letzten Jahren hat besonders eine Form der Algorithmen eine Renaissance[9] erlebt: *Algorithmen, die auf neuronalen Netzen basieren.* Ob diese die Quelle der intelligenten Algorithmen ist, soll daher folgend betrachtet werden.

Neuronale Netze stellen die Basis für einen neuartigen Typ von Algorithmen. Während musterbasierte Algorithmen noch vordefinierte Muster benötigten, die sie anschließend in den Daten zu finden versuchen, sind neuronale Netze selbst in der Lage, Muster zu generieren und zu finden, ganz ohne Hilfe. Sie agieren selbstlernend. Damit werden diese immer unabhängiger darin, Aufgaben zu erledigen. Die Idee der neuronalen Netze weicht von der deterministischen Grundidee ab und basiert dabei auf uns selbst. Die Grundlage liegt in der Gehirnforschung. Dabei nutzt diese die gewonnenen Erkenntnisse über das gegenseitige Spiel der *Nervenzellen (Neuronen)* und deren *Verbindungen (Synapsen)* in unserem Gehirn. Hunderte und tausende Verbindungen finden wir für ein Neuron mit anderen. Diese sind flexibel und höchst wandelbar. Die Kommunikation zwischen den einzelnen Neuronen erfolgt in der Regel mittels chemischer Botenstoffe (sogenannte Neurotransmitter). Innerhalb von Neuronen werden Signale elektrisch weitergeleitet. Die Stärke des Einflusses auf eine Nervenzelle hängt davon ab, wie nah eine Synapse am Zellkörper ansetzt. Bei gleichzeitig eintreffenden Signalen unterschiedlicher Nervenzellen addieren sich die Wirkungen. Die Botenstoffe lösen dann in einem Neuron einen sich immer weiter verstärkenden Reiz aus. Erreicht der Reiz eine bestimmte Schwelle, dann wird ein Aktionspotenzial ausgelöst. Dieses komplexe Geflecht aus Neuronen, Synapsen und Aktionspotenzial ist hochgradig veränderbar – durch Erfahrungen und Eindrücke finden fortwährend Anpassungen statt. Wichtig ist, dass sich spezifische Gruppierungen einzelner Neuronen bilden und sich auf das Erkennen komplexer Muster spezialisieren. Beispielsweise Gesichter: es sind unterschiedliche Neuronenverbände beteiligt, wenn wir einfach Gesichter im allgemeinen oder Gesichter unserer Freunde erkennen.

[9] Im Jahr 1969 zeigten Minsky und Papert, dass viele Probleme mit den damals bestehenden neuronalen Netzwerken gar nicht gelöst werden können. Damit wurde es still um diesen Forschungsbereich. Erst 15 Jahre später erlebte das Gebiet mit vielversprechenden Arbeiten u.a. von Hopfield (1985) eine Renaissance, welche bis heute ungebrochen ist (siehe [Kr07, p. 9ff.]).

Die technische Adaption der Gehirnforschung hin zu neuronalen Netzen macht sich diese Eigenschaften zunutze. Es geht in erster Linie um das Erkennen und Erlernen von Mustern. Die neuronalen Netze sind in ihren Grundelementen von hoher Simplizität. Sie setzen sich aus einfachen miteinander gekoppelter mathematischer Funktionen zusammen. Mit deren Hilfe diese Netze erst im Laufe des Trainings und nach der Betrachtung einer Vielzahl von Beispielen in die Lage versetzt werden, das zu lernen, was wichtig ist (z.B. was ein Gesicht zu einem Gesicht macht oder einen Hund zu einem Hund). Jedes neue Beispiel nutzt das Netzwerk dabei zur Aktualisierung der Muster und speist dies als Feedback zurück in das neuronale Netzwerk, in dem es unzählige Paramater innerhalb des Netzwerkes neu justiert. Dadurch passen sich die elementaren Grundfunktionen im Verbund über die Zeit an die Beispiele und werden in die Lage versetzt, Aufgaben zu lösen bzw. Probleme zu bearbeiten. Die Anpassung ist dabei Domain ungebunden und kann in beliebig vielen Lernschritten wiederholt werden, damit es immer besser die Aufgabe lösen kann. Wird ein Bild ins Netzwerk gespeist, dann lernt es beispielsweise die Muster eines Tieres, gleiches gilt für einen Satz, auch wenn uns Bilder und Sprache erst einmal recht unterschiedlich in der Verarbeitung erscheinen.

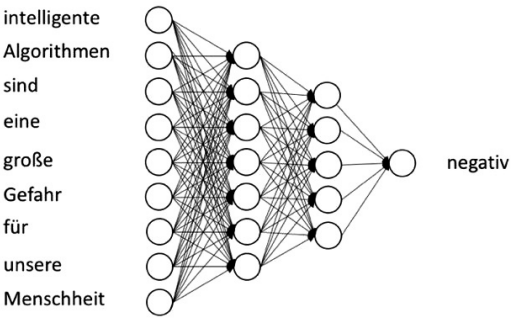


Abb. 3: Neuronales Netzwerk lernt Muster eines negativen Satzes (eigene Darstellung)

Um die Vorgehensweise eines neuronalen Netzwerkes zu verdeutlichen, soll das Beispiel aus den musterbasierten Algorithmen wieder aufgegriffen werden. Soll für einen gegebenen Satz entschieden werden, ob dieser positiv oder negativ ist, dann bekommt das neuronale Netzwerk diesen als Eingabe (siehe Abb. 3). Da diese Form nicht mit Buchstaben umgehen kann, wird jedes Wort in eine Zahlenfolge übersetzt z.B. die Stelle im Wörterbuch. Diese Zahlen werden vorbei an vielen kleinen mathematischen Stellschrauben durch das Netzwerk geleitet. Am Ende entscheidet das Netzwerk, ob es sich um einen positiven oder negativen Satz handelt. Liegt es falsch, dann ändert es seine Stellschrauben, um beim nächsten Satz dieser Art richtig zu entscheiden. Diese Stellschrauben bilden musterartig ab, was es heißt, wenn Sätze eine positive oder negative Form haben.

Diese neuartigen Algorithmen sind damit um eine neue Komplexitätsstufe erweitert, aber sind sie damit auch intelligent? Als erstes kann beobachtet werden, dass diese nicht gewohnheitsmäßig handeln. Ihre Entscheidungen sind kein Abklatsch und sie sind in der Lage, Neuerungen aufzunehmen. Dies bedeutet auch, dass sie aus Fehlern lernen können.

Passt das Netzwerk nicht zu der Aufgabe, dann verändert sich der (selbstlernende) Algorithmus selbstständig[10]. Er modelliert damit die Wirklichkeit anhand von Erfahrung. Ebenfalls wird deutlich, dass ein Netzwerk, was bereits Muster gelernt hat, nicht aus Zufälligkeit, sondern auf der Basis der Vorgänger agiert. Innerhalb des Netzwerkes ist eine starke logikbasierte Vorgehensweise vorzufinden. Das Netzwerk lernt anhand von mathematischen Funktionen und definierten Verknüpfungen. Dennoch muss hier herausgestellt werden, dass auch das Netzwerk nur in seinem vorgegebenen Horizont lernt und nicht davon losgelöst eine Fähigkeit besitzt. Diese Algorithmen sind in ihrem aktuellen Einsatz auch von ihrer strukturellen Definiertheit durch den Entwickler bestimmt. Dies bedeutet, wenn ein Algorithmus merkt, dass er falsch liegt, dann kann er seine internen Netzwerkparameter ändern, er ist aber nicht in der Lage, sich lernend selbst ein neues Netzwerk zu bauen und sich dahingehend selbst strukturell zu verändern[11]. Womit auch für die dargestellte aktuelle Generation von Algorithmen keine vollständige Intelligenz nach Gilbert Ryles Intelligenzbegriff postuliert werden kann. Gleichwohl eine Tendenz zur Erweiterung intelligenter Elemente deutlich erkennbar ist.

5 Schlussfolgerung

Intelligente Algorithmen im Bereich der künstlichen Intelligenz sind ein allgegenwärtiges Thema, sowohl in der Forschung, als auch in der Öffentlichkeit. Die mediale Darstellung berichtet von vollständig autonomen und intelligenten Maschinen. Ob Algorithmen, aus denen sich (digitale) Maschinen zusammensetzen, bereits eine Form der Intelligenz haben, wurde von mir in dieser Arbeit betrachtet.

Dazu ging ich einführend auf das Verständnis einer intelligenten Handlung nach Gilbert Ryle (1969) ein. Nach diesem ist eine Handlung intelligent, wenn sie folgende Eigenschaften besitzt. Die Handlung darf *nicht gewohnheitsmäßig* sein, sie muss *absichtsvoll* und *nicht aus Zufall* passieren, sie muss *logischen Gesetzmäßigkeiten* folgen und intelligent gehandelt werden kann nur, wenn aus *Fehlern gelernt* wird. Anschließend wurde ein Verständnis des Begriffs Algorithmus erarbeitet. Dieser wurde als vordefinierte Handlungsanweisung zur Problemlösung beschrieben. Im Verlauf wurde deutlich, dass dies der heutigen Verwendung von Algorithmen nicht mehr gerecht wird, da diese bereits vielfältig ausdifferenziert wurden und um einige Komplexitätsstufen gewachsen sind. Dies führte uns zu merkmalsbasierten Algorithmen, die in der Lage sind, anhand fest definierter Merkmale beliebig neue Daten zu betrachten. Fortführend wurde auf die heute vorherrschenden Algorithmen, in Form von neuronalen Netzen, eingegangen. Diese sind es, welche heute im Bereich der künstlichen Intelligenz verstärkt Anwendung finden.

[10] Hier sei angemerkt, dass sich dies im Besonderen auf das Lernen der Gewichtungen bezieht. Die Art der Problemlösung, der Gewichtungsprozess sowie die Bewertung eines Fehlers bleiben gleich.
[11] Hierfür bedürfte es bspw. einer Anknüpfung an die Idee des *self-modifying code* [Ba81].

Bei der Untersuchung der verschiedenen Typen von Algorithmen wurde deutlich, wie stark sich die Beschaffenheit der Algorithmen auf die Betrachtung des Intelligenzverständnisses auswirkt. Die einfachen Problemlöser, die traditionellen Algorithmen, zeigen durch ihre vorgedachte statische Struktur, dass diese gewohnheitsmäßig reproduzieren, aber nicht in der Lage sind, aus Fehlern zu lernen. Dies erfüllte vollumfänglich nicht die Anforderungen, die Ryle an die intelligente Handlung stellt. Bei der Betrachtung des merkmalsbasierten Algorithmus zeigte sich zum einen, dass er logisch, auf statistischer Basis, nicht zufällig Handlungsentscheidungen trifft und nicht gewohnheitsmäßig auf neue Situationen reagieren kann. Zum anderen zeigt sich jedoch auch, dass diese Form durch vorprogrammierte Muster nur einen begrenzten Horizont hat, welcher durch eine strukturelle Unveränderbarkeit bestimmt ist. Er ist zwar in der Lage, auf Basis von vielen Erfahrungen zu lernen, aber er kann daraus keine Veränderung des eigentlichen Vorgehens erzeugen. Er ist nicht in der Lage, selbst neue Muster zu bestimmen, um seine Aufgabe noch besser zu erfüllen. Auch dieser Algorithmus stellte sich als nicht intelligent heraus. Diesen beiden Algorithmen entgegen verhielt es sich bei den Algorithmen, die auf neuronalen Netzwerken basieren. Diese können effektiv aus Fehlern lernen und ihre Netzwerkeigenschaften darauf anpassen. Damit begegnen diese Algorithmen mühelos Neuheiten und sind weder gewohnheitsmäßig, noch zufällig bestimmt. Doch auch dieses Netzwerk kann nur in einem vordefinierten Rahmen lernen. Es ist zwar möglich, dass das Netzwerk seine internen Parameter justiert, es kann sich jedoch selbst als Gesamtes nicht lernend umbauen oder seine Struktur reflektieren. Damit zeigte die aktuellste Form von Algorithmen zwar eine viel höhere Komplexitätsstufe und erfüllte eine Vielzahl der Anforderungen an einen intelligenten Algorithmus, kann jedoch auch abschließend nicht als intelligent bezeichnet werden. Damit muss folgernd die Forschungsfrage verneint werden: *Algorithmen sind (noch) nicht intelligent!*

Abschließend bleibt festzuhalten, um wirklich intelligent sein zu können, müssen Algorithmen lernend die Grenzen ihrer eigenen Verfasstheit überwinden. Sie selbst müssen zum intelligenten Konstrukteur werden. Diesen Pfad auch kritisch zu betrachten, wie es Dirk Helbing und viele andere tun, sollte dabei ein integraler Bestandteil sein. Kriterien, wie die von Gilbert Ryle, können dabei einen wichtigen Beitrag leisten, um die Facetten der Intelligenz auszuloten und zu hinterfragen.

Literaturverzeichnis

[An17] Anderl, S.: Künstliche Intelligenz, http://www.faz.net/aktuell/feuilleton/debatten/die-risiken-kuenstlicher-intelligenz 15163407.html, 2017, Stand: 29.04.2018.

[Ba81] Bashe, C./ Buchholz, W./ Hawkins, G./ Ingram, J./ Rochester, N.: The Architecture of IBM's Early Computers. In: IBM Journal of System Development, 25(5), 1981.

[Be16] Betschon, S.: Versteckspiel im Suchbaum, www.nzz.ch/meinung/kommentare/versteckspiele-im-suchbaum-1.18717802, 2016, Stand: 29.04.2018.

[Bu17] Budras, C.: Künstliche Intelligenz, www.faz.net/aktuell/wirtschaft/netzwirtschaft/kuenstliche-intelligenz-der-grosse-durchbruch-ist-jetzt-da-15141071-p2.html, Stand am 29.04.2018.

[Ca17] Caracciolo, L.: Grundsatzfragen der künstlichen Intelligenz. Können wir den Maschinen noch trauen? In: t3n Magazin Nr. 48, S. 76-79, 2017.

[Da50] Dayton, M. C.: Logistic Regression Analysis, https://www.researchgate.net/publication/268416984, 1950, Stand: 29.04.2018.

[Gr15] Graff, B.: Deep Dream: Ein erschreckend intelligenter Algorithmus, http://www.sueddeutsche.de/digital/kuenstliche-intelligenz-hilfe-die-computer-bekommen-augen-1.2570782, Stand: 29.04.2018.

[He15] Helbing, D./ Frey, B. S./ Gigerenzer, G./ Hafen, E./ Hagner, M./ Hofstetter, Y./ Van den Hoven, J./ Zicari, R. V./ Zwitter, A.: Digitale Demokratie statt Datendiktatur, http://www.spektrum.de/news/wie-algorithmen-und-big-data-unsere-zukunft-bestimmen/1375933, 2015. Stand: 29.04.2018.

[He16] Helbing, D.: Maschinelle Intelligenz – Fluch oder Segen, https://www.telekom.com/de/konzern/digitale-verantwortung/details/maschinelle-intelligenz---fluch-oder-segen--es-liegt-an-uns---352200, 2015. Stand: 29.04.2018.

[Ke75] Kemmerling, A.: Gilbert Ryle: Können und Wissen. In: Speck, Josef (Hrsg.): Grundprobleme der großen Philosophen – Philosophie der Gegenwart III, Göttingen, Vandenhoeck & Ruprecht, S. 127-167, 1975.

[Kr07] Kriesel, D.: Ein kleiner Überblick über neuronale Netze. http://www.dkriesel.com/_media/science/neuronalenetze-de-zeta2-2col-dkrieselcom.pdf, 2007. Stand: 29.04.2018.

[KWM11] Kouloumpis, E./ Wilson, T./ Moore, J.: Twitter Sentiment Analysis: The Good the Bad and the OMG! In: Proceedings of the Fifth International AAAI Conference on Weblogs and Social Media, S. 538-541, 2011.

[Mc07] McCarthy, J.: What is artifical intelligence? http://www-formal.stanford.edu/jmc/whatisai.pdf, 2007. Stand: 29.04.2018.

[PD08] Pomberger, G./ Dobler, H.: Algorithmen und Datenstrukturen. Eine systematische Einführung in die Programmierung, München, Pearson Studium, 2008.

[Po00] Popper, K.: Wissenschaft: Vermutungen und Widerlegungen, Vortrag in Cambridge, Tübingen, Mohr Siebeck, 2000.

[Ry69] Ryle, G.: Der Begriff des Geistes, Stuttgart, Reclam, 1969.

[So13] Socher, R./ Perelygin, A./ Wu, J./ Chuang, J./ Manning, C./ Ng, A./ Potts, C.: Recursive deep models for semantic compositionality over a sentiment treebank. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, S. 1631-1642, 2013.

[Wo15] Wolfangel, E.: Maschinen mit gottesgleichem Wissen, https://www.stuttgarter-zeitung.de/inhalt.big-data-maschinen-mit-gottesgleichem-wissen.12c1211e-c88c-4aa3-959a-d4ed913fcc8e.html, 2015. Stand: 29.04.2018.

Auswirkung von Veränderungen des geomagnetischen Felds auf Migräneanfälle

Noah Lankl[1], Marvin Kirsch[2] und Felix Wünsche[3]

Abstract: Innerhalb dieses Papers werden Veränderungen des geomagnetischen Felds in Verbindung mit dem Auftreten von Migräneanfällen von circa 6000 Patienten untersucht. Ziel ist es, eine Aussage darüber zu treffen, ob genannte Änderungen einen Einfluss auf Migräneanfälle haben. Als Basis für diese Untersuchung werden zunächst die Daten vorverarbeitet. Im Anschluss folgt die Analyse. Anschließend wird das Vorgehen kurz diskutiert und schlussendlich ein Fazit gezogen. Unsere Ergebnisse deuten auf einen statistisch signifikanten Einfluss des geomagnetischen Felds auf Migräneanfälle hin.

Keywords: Migräneanfälle, Migräne, geomagnetische Veränderungen, Korrelation

1 Motivation

Die Auslöser von Migräneanfällen sind auch heute noch nicht ausreichend erforscht. Dieser Ursache ist es geschuldet, dass Betroffene sich häufig mit möglichen Auslösern der eigenen Migräneanfälle beschäftigen und infolge der Erkrankung in ihrer Lebensweise und ihrem täglichen Tagesablauf beeinträchtigt werden. So stellen sich beispielsweise Fragen wie: "Habe ich falsche Lebensmittel zu mir genommen?", "Lege ich ein falsches Verhalten an den Tag?", "Ist das Wetter oder Wetterwechsel schuld an meinen Migräneanfällen?" und vor allem "Kann ich irgendetwas tun, um Migräneanfällen vorzubeugen?".

Auch Änderungen des geomagnetischen Felds stehen mitunter im Verdacht, Migräneanfälle auszulösen. Aufgrund dessen wird innerhalb dieses Papers die Veränderung des geomagnetischen Felds als möglicher Auslöser von Migräneanfällen untersucht. Ähnliche Arbeiten wurden bereits in den 1980er- und 1990er-Jahren durchgeführt. Dabei unterstützen manche Studien unsere Ergebnisse [Ma94], [St99]. Wobei sich andere nur mit Absolutwerten auseinandersetzen und deren Anstieg eine erhöhte starke Kopfschmerzanzahl korreliert wurde [Ku87]. Anreiz für diese Arbeit

[1] Hochschule für Angewandte Wissenschaften Hof, Alfons-Goppel-Platz 1, 95028 Hof, nlankl@hof-university.de
[2] Institut für Informationssysteme der Hochschule für Angewandte Wissenschaften Hof, Forschungsgruppe Analytische Informationssysteme, Alfons-Goppel-Platz 1, 95028 Hof, mkirsch@hof-university.de
[3] Institut für Informationssysteme der Hochschule für Angewandte Wissenschaften Hof, Forschungsgruppe Analytische Informationssysteme, Alfons-Goppel-Platz 1, 95028 Hof, fwuensche@hof-university.de

lieferten uns die geringen Patientenzahlen von circa 30-40, wohingegen für diese Untersuchung ungefähr 6000 Patienten zur Verfügung standen.

Zur Analyse der Daten wird ein statistisches Hilfsmittel benötigt. Hierfür wurde in diesem Fall die statistische Programmiersprache R verwendet. Dies lässt sich folgendermaßen begründen. Zunächst handelt es im wissenschaftlichen Umfeld um eine vielfach verwendete Programmiersprache. Des Weiteren wird diese ebenfalls in der Forschungsgruppe Analytische Informationssysteme verwendet. Außerdem konnten teilweise bereits vorhandene R-Skripte in dieser Arbeit verwendet werden.

2 Datenvorverarbeitung

Zur Analyse der Korrelation zwischen geomagnetischen Veränderungen und Migräneanfällen müssen die vorliegenden Daten vorverarbeitet werden.

2.1 Ausgangsdaten

Migräne-Radar: Als Basis für die durchgeführten Untersuchungen dienen freiwillig gemeldete Migräneanfälle von circa 6.000 Patienten. Diese haben in einem Zeitraum von zwei Jahren circa 75.000 Migräneanfälle im Projekt Mira (Migräne-Radar) der Forschungsgruppe Analytische Informationssysteme am iisys (Institut für Informationssysteme der Hochschule für Angewandte Wissenschaften Hof) seit Juni 2015 gemeldet. Obwohl Anfälle auch rückwirkend gemeldet werden können, werden für die hier getätigten Analysen Anfälle ab Juni 2015 bis Ende Oktober 2017 betrachtet. Dies begründet sich durch den offiziellen Start des Forschungsprojekts und der wenigen und somit nicht repräsentativen Meldungsdichte vor diesem Zeitraum. Des Weiteren muss eine weitere Filterung dieses Datensatzes durchgeführt werden um anschließend korrekte Analyseergebnisse liefern zu können:

- Entfernen von chronischen Migräne-Patienten, da hier fraglich ist, ob deren Migräne mit äußeren Einflüssen zusammenhängt
- Entfernen von Patienten, die weniger als 10 Anfälle gemeldet haben, da diese nur weniger genaue Analyse zulassen

Für die durchzuführenden Analysen stehen nach der Filterung circa 25.000 Anfälle zur Verfügung.

Geomagnetische Messdaten: Weltweit existieren 13 verteilte Messstationen, mit Hilfe derer Messdaten das geomagnetische Feld bestimmt wird. Die Störungen des geomagnetischen Felds durch solare Partikelstrahlung werden von einem Magnetometer in nano Tesla (nT) erfasst. Aus dem höchsten Störungswert innerhalb eines Dreistunden-Intervalls wird ein K-Index als ganze Zahl von 0-9 gebildet. Mit 1 für ruhig und 5 oder

mehr als Indiz für einen geomagnetsicher Sturm. Das K steht hierbei für Kennziffer. Aus dem *K*-Index lassen sich folgende weitere Werte ableiten [Iv97]:

- *Kp (planetarische Kennziffer):* mittlerer Wert aus dem Störungsgrad K an den 13 Stationen
- *Ap* Ableitung von Kp-Index nach festgelegter Tabelle
- *Ap* bezeichnet den Mittelwert der acht *Ap*-Werte, welche sich auf einen Tag verteilen.

Innerhalb der in dieser Arbeit getätigten Analysen werden primär die *Ap*-Werte beziehungsweise deren Mittelwert (*Ap*) verwendet. Da sich alle *Ap*-Werte aus den *Kp*-Werten direkt ableiten lassen, wäre prinzipiell auch eine Verwendung der *Kp*-Werte möglich.

Datenformat: Zur Korrelation der vorliegenden Migräneanfälle mit Veränderungen des geomagnetischen Felds werden die vom deutschen Geo-Forschungszentrums (GFZ) veröffentlichten Kp-Indizes verwendet.

Die hieraus erhaltenen Daten wurden bereits in der Forschungsgruppe Analytische Informationssysteme am iisys entsprechend vorverarbeitet, sodass eine .csv (comma seperated values) Datei in folgendem Format vorliegt:

2.2 Vorverarbeitung der geomagnetischen Messdaten

Da alle geomagnetischen Messwerte gesammelt in einer .csv-Datei im zuvor beschriebenen Format vorliegen, müssen diese zunächst für den Verwendungszweck angepasst werden.

Programmfunktionalität: Um die weitere Analyse mit R einfacher zu gestalten, wurde ein Java-Programm geschrieben, welches die jeweilige Änderung des Ap-Messwertes zum Vortag mit dem zugehörigen Datum in eine .csv Datei schreibt. Die Daten liegen daraufhin in folgendem Format vor:

Listing 1: Exportiertes Format

Ap_diff | date

Beispiel 2: Exportierte Werte

6 | 02.06.2015

Ap_diff stellt hierbei die Veränderung im *Ap*-Messwert zum Vortag auf den in der exportierten Datei vermerkten Tag dar.

Im gezeigten Beispiel gab es demnach eine Änderung im *Ap*-Wert vom 01.06.2015 auf den 02.06.2015 um 6 Einheiten im positiven Bereich. Sollten Ap-Werte sinken, werden die Daten entsprechend mit negativem Vorzeichen durch das Programm exportiert.

2.3 Vorverarbeitung der Migräne-Daten

Der Beispieldatensatz von Migräneanfällen liegt für die hier getätigten Analysen bereits vorgefiltert vor. Dieser Datensatz stammt ebenfalls aus der Forschungsgruppe Analytische Informationssysteme am iisys. Demnach müssen für die Migräne-Daten keine weiteren Vorverarbeitungen mehr getätigt werden.

Ein Datensatz für Migräneanfälle ist wie folgt aufgebaut:

- Patienten-Referenz
- Id
- Anfangszeit
- Endzeit
- Weitere Daten, die für die hier getätigten Analysen nicht verwendet wurden

3 Analyse

3.1 Vorgehensweise

Zunächst wurde die Fragestellung festgelegt, die in dieser Arbeit untersucht wird. Untersucht werden soll demnach eine mögliche Korrelation zwischen Änderungen des geomagnetischen Felds und der Anzahl von Migräneanfällen.

Um die Häufigkeit von Migräneanfällen mit geomagnetischen Veränderungen zu korrelieren, muss diverse Vorarbeit geleistet werden. So müssen Daten wie Patientenzahlen und Häufigkeit von Ap-Wert-Veränderungen normalisiert werden, um eine gemeinsame Vergleichsbasis für die späteren Analysen zu schaffen. Das genaue Vorgehen zur Normalisierung wird in den folgenden Kapiteln beschrieben.

Anschließend werden Ap-Wert-Änderungen mit der Häufigkeit von Migräneanfällen korreliert, sodass sich neben der statistischen Beweisführung ebenfalls eine aussagekräftige Visualisierung erzeugen lässt.

3.2 Parameter zur Datenanalyse

Geomagnetische Messdaten: Die geomagnetischen Messdaten bilden neben den Migräneanfällen die Basis der durchzuführenden Analysen. Es werden die Änderungen der geomagnetischen Messwerte an jedem Tag bezüglich zum Vortag bestimmt. Die hierbei errechnete Verteilung wird wie folgt dargestellt:

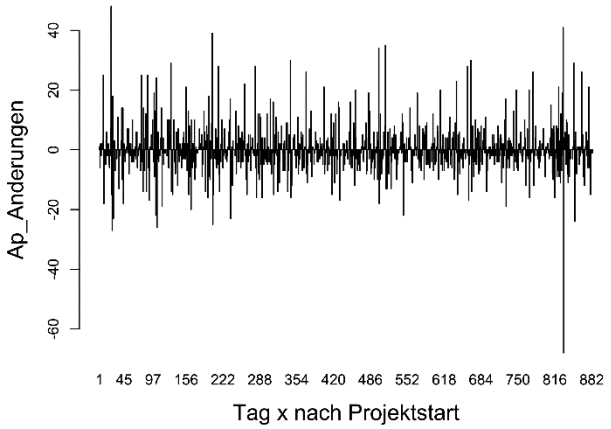


Abb. 1: Verlauf Ap-Messwert-Veränderungen 06.2015 bis 10.2017

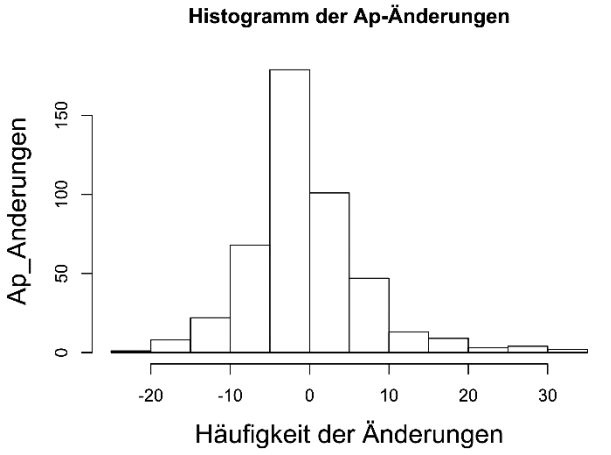


Abb. 2: Histogramm Auftreten Ap-Messwert-Veränderungen

Gemeldete Migräneanfälle: Da durch das Meldeverfahren des Migräne-Radar nicht bekannt ist, ob ein Patient aktuell nicht mehr am Projekt teilnimmt oder lediglich keine Migräneanfälle erleidet, müssen die vorliegenden Anfälle zunächst normalisiert werden beziehungsweise die Anzahl der potentiell teilnehmenden Patienten an einem Tag berechnet werden. Zunächst wird jedoch der Verlauf der nicht normalisierten Anfälle betrachtet.

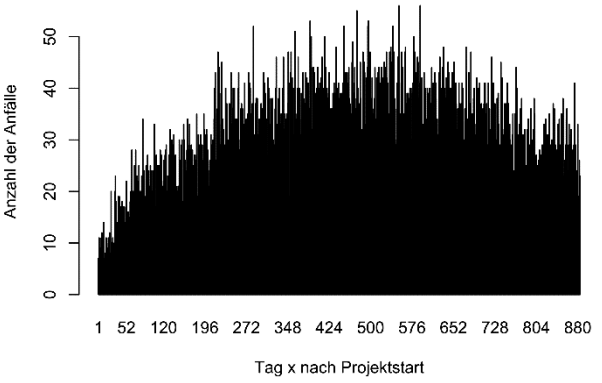


Abb. 3: Anzahl der Anfälle nicht normalisiert auf potentielle Patienten

Zu sehen ist eine steigende Anzahl von meldenden Patienten bis zu einem Höhepunkt zum Ende des Jahres 2016. Aktuell ist eine sinkende Tendenz in der Anzahl der meldenden Patienten zu erkennen.

Bei der Normalisierung wird wie folgt vorgegangen:

Für jeden Patienten wird dessen erster und letzter Anfalltag ermittelt. Innerhalb dieser Zeitspanne wird der Patient als aktiv betrachtet und damit zu den Teilnehmern in diesem Zeitraum addiert.

Für jeden Tag wird die Anzahl der Anfälle durch die kumulierten aktiven Patienten in dieser Zeitspanne dividiert. Das Resultat ist in Abbildung 5 zu sehen.

Hierdurch erhält man für jeden Tag eine normalisierte Anzahl von Anfällen, mit der weitere Analysen getätigt werden können.

Der Verlauf der normalisierten Anzahl von Anfällen stellt sich wie folgt dar:

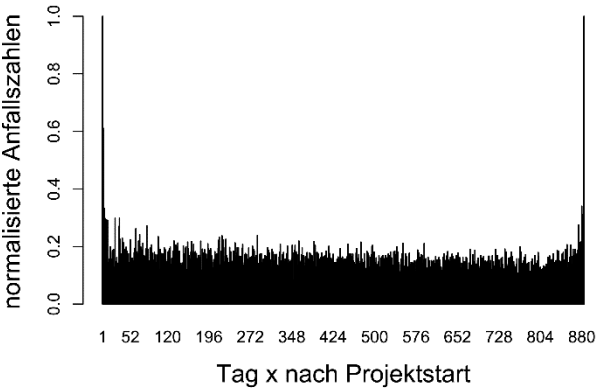


Abb. 4: Anzahl der Anfälle normalisiert auf Patienten

Nach der Normalisierung ergibt sich erwartungsgemäß eine relativ gleichmäßig verteilte Anzahl an potentiell teilnehmenden Patienten. Der Anstieg der normalisierten Anfälle auf den Wert Eins an den Enden der Zeitpanne lässt sich dadurch begründen, dass die Anzahl der Patienten mit den gemeldeten Anfällen pro Tag übereinstimmt. Ansonsten meldet im Schnitt jeder fünfte teilnehmende Patient pro Tag in etwa einen Anfall.

3.3 Durchführung

Datenanalyse: Nach der bisherigen Vorverarbeitung der Daten ist nun die eigentliche Analyse in R möglich. Dabei wird wie folgt vorgegangen:

Zunächst werden die für die Analyse benötigten Daten in R eingelesen. Dabei handelt es sich um die zuvor erwähnten Dateien, welche die Ap-Abweichungen und die Anfälle mit Patientendaten enthalten. Um diese in R vergleichbar zu machen, werden im nächsten Schritt zunächst alle Datumswerte auf dasselbe Format gebracht.

Anschließend werden mithilfe der Anfallsdaten des Migräne-Radars und der Tabelle der Ap-Abweichungen die Anzahl der Anfälle, auf Ap-Änderungen zugeordnet, aufsummiert. Hierbei ist zu beachten, dass von einer bestimmten Anzahl an Tagen nach einer Ap-Änderung als Auslöser für Migräneanfälle ausgegangen wird. Für die Analyse in dieser Arbeit wurde von einem Auslöser einen Tag vor den jeweiligen Anfällen ausgegangen. Dies lässt sich aus Erfahrungswerten des Forscherteams im Institut für Informationssysteme begründen. Für einen jeweiligen Ap-Wert werden die normalisierten Anfälle für diesen Ap-Wert aufsummiert.

Um ein korrektes Analyseergebnis zu erreichen, wird gleichzeitig die Häufigkeit des Vorkommens jeder Ap-Änderung mitgezählt. Mithilfe dieser Daten werden anschließend die zuvor berechneten Summen von Anfällen pro Ap-Änderung normalisiert.

Als Ergebnis dieses Vorgehens ergibt sich ein Vektor mit den für jede Ap-Änderung aufsummierten und normalisierten Anfällen.

Berechnung von Fehlerbalken: Da kleine Änderungen der Ap-Werte in der Analyse häufig auftreten, wohingegen große Änderungen relativ selten vorkommen, entsteht hier die Gefahr einer Fehlanalyse, da somit für die seltenen, großen Änderungen nur wenige Anfallstage zur Verfügung stehen. Um aussagekräftige Ergebnisse zu erhalten, ist es deshalb nötig, für die Summen der Anfälle pro Ap-Änderung jeweils den statistischen Fehler zu berechnen und mithilfe von Fehlerbalken darzustellen. Über diese können anschließend sowohl über den positiven als auch über den negativen Bereich der Ap-Änderungen Regressionsgeraden mit minimalem Abstand zu den einzelnen Punkten des Plots angelegt werden, wobei Punkte mit großen Fehlerbalken weniger in die Berechnung einfließen als Punkte mit kleinen Fehlerbalken.

Berechnung der Regressionsgeraden: Um einen möglichen Anstieg der normalisierten Anfallszahlen zu visualisieren, empfiehlt sich die Berechnung von Regressionsgeraden, dies sowohl für negative als auch für positive Ap-Änderungen.

Zunächst müssen zur Berechnung der Geraden folgende Vorfaktoren ermittelt werden [Le94]:

$$\mathrm{A} = \sum \frac{x_i}{\sigma_i^2}, B = \sum \frac{1}{\sigma_i^2}$$
$$C = \sum \frac{y_i}{\sigma_i^2}, D = \sum \frac{x^2}{\sigma_i^2}$$
$$\mathrm{E} = \sum \frac{x_i y_i}{\sigma_i^2}, \mathrm{F} = \sum \frac{y_i^2}{\sigma_i^2}$$

Im Anschluss daran kann daraus die Steigung der Geraden a und der y-Achsen-Abschnitt b berechnet werden [Le94]:

$$a = \frac{EB - CA}{DB - A^2}$$
$$b = \frac{DC - EA}{DB - A^2}$$

Somit lässt sich die Geradengleichung $y = ax + b$ aufstellen. Der zugehörige Plot ist in Abbildung 7 zu sehen.

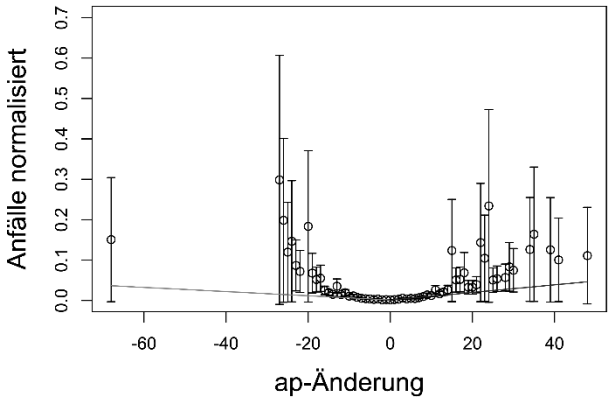


Abb. 5: Anzahl der Anfälle normalisiert auf potentielle Patienten und Ap-Häufigkeit mit Fehlerbalken und Regressionsgeraden

Auswertung des finalen Plots: Wie in diesem Plot zu erkennen ist, sind die berechneten Fehlerbalken für kleine Ap-Änderungen im Bereich von -20 bis 0 beziehungsweise von 0 bis +20 sehr klein. Die Ursache hierfür ist, dass kleine Änderungen verhältnismäßig oft vorkommen und dementsprechend viele Anfallstage existieren. Für seltene, betragsmäßig große Ap-Änderungen existieren demnach deutlich weniger Anfallstage, woraus deren große Fehlerbalken resultieren.

Bei Betrachtung der entstandenen Regressionsgeraden fällt zunächst auf, dass diese eine Steigung zu den hohen Ap-Änderungen aufweisen. Hieraus könnte der falsche Schluss gezogen werden, dass sich hohe Ap-Änderungen tatsächlich auf Migräneanfälle auswirken. Dies lässt sich hieraus allerdings noch nicht schließen. Somit muss zuletzt die Signifikanz der Regressionsgeradensteigungen überprüft werden. Hierfür werden 2 Methoden kurz angeführt. Zunächst wird eine Faustregel betrachtet, die besagt, dass die Geradensteigung a am besten drei oder mehr mittlere Standardfehler grösser als 0 sein sollte. Der mittlere Standardfehler der Regressionsgeraden lässt sich wie folgt berechnen [Le94]:

$$\sqrt{\sigma^2(a)} = \sqrt{\frac{B}{BD - A^2}}$$

Teilt man den jeweiligen Wert durch die jeweilige der Geradensteigung, ergibt sich für beide Geraden ein Wert von circa 8, was bedeutet, dass beide Geraden eine statistisch signifikante Steigung aufweisen.

Genauer arbeitet der t-Test: Hier wird der berechnete Wert des Regressionskoeffizienten in Beziehung zu seinem Standardfehler gesetzt. Dieser bestätigte das obige Ergebnis.

Die hier durchgeführten Analysen wurden exemplarisch auf den zum Zeitpunkt der Arbeit vorhandenen Daten ausgeführt. Prinzipiell lassen sich jedoch auch andere Datensätze mit Migräneanfällen mithilfe der vorgestellten Methodik untersuchen, sofern diese auf dieselbe Weise wie hier beschrieben vorgefiltert werden und im selben Format vorliegen.

3.4 Interpretation

Aus dem finalen Plot kann man eine Korrelation zwischen geomagnetischen Veränderungen und Migräneanfällen schließen. Diese kann jedoch darauf basieren, dass die verwendeten Daten diese zufällig hervorgerufen haben. Dies könnte mit Hilfe zufällig generierter Anfälle nachgewiesen werden. Bei der Auswertung dieser dürfte der Anstieg der normalisierten Anfälle bei hohen Ap-Wert-Änderungen nicht vorhanden sein.

Ein definitiv unbestreitbarer kausaler Zusammenhang ist selbst bei sehr vielen Migräneanfällen über eine lange Zeitspanne mit dieser Vorgehensweise nicht nachweisbar. Dies lässt sich dadurch begründen, dass der Faktor, der die signifikante Geradensteigung auslöst, nicht sicher ermittelt werden kann.

4 Diskussion

Im Folgenden wird die Vorgehensweise zur Durchführung des Projekts kurz diskutiert. Zunächst könnte man den gewählten Ansatz hinterfragen.

Eine andere Herangehensweise wäre die Berechnung signifikanter Ap-Wert-Änderungen über die Standardabweichung von der Median-Ap-Wert-Änderung. So hätte man mithilfe eines t-Tests auf normalisierte Anfälle bei statistisch signifikanten Ap-Wert-Änderungen vergleichen können. Die hier verwendete Herangehensweise wurde gewählt, da so alle normalisierten Anfälle bei allen Ap-Änderungen in einem Graphen sichtbar werden.

Ein weiterer Vorteil der Vorgehensweise ist, dass mit dieser nahezu jede beliebige Ursache mit Migräneanfällen korreliert werden kann.

Da nur ein Anfall pro Tag gemeldet werden kann, wurde auf den Ap-Wert eines Tages zur Analyse zurückgegriffen.

5 Fazit

Die hier erarbeiteten Ergebnisse lassen für sich selbst betrachtet noch keine Schlussfolgerungen auf einen kausalen Zusammenhang von geomagnetischen Veränderungen mit Migräneanfällen zu. Es mag auf den ersten Blick als ungewöhnlich erscheinen, die geomagnetischen Veränderungen mit den Migräneanfällen zu korrelieren, da dies als Faktor bei bisherig bekannteren Analysen meist außer Acht geblieben ist.

Jedoch müssen auch unbekanntere Faktoren bei der Suche nach Migräneauslösern betrachtet werden, weil selbst der Ausschluss von Faktoren bei der Ursachenforschung von Migräne hilfreich sein kann.

Zukünftig werden innerhalb der Forschungsgruppe Analytische Informationssysteme am iisys der Hochschule für Angewandte Wissenschaften Hof weitere Analysen auf Basis der vorgestellten Methodik weitergeführt. Hierfür kann diese Arbeit hilfreiche Ansätze liefern.

Literaturverzeichnis

[Iv97] Ivory K., Geomagnetic Ap, Ap, Cp, and C9 Indices, 1997, 13.02.2018

[Ku87] A. Kuritzky M.D. et al., Headache, https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1526-4610.1987.hed2702087.x, Stand: 24.06.18.

[Le94] William R. Leo. Techniques for Nuclear and Particle Physics Experiments. Springer, Berlin. 2 nd, rev. ed. 1994

[Ma94] G. De Matteis, 1994, Headache, https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1526-4610.1994.hed3401041.x, Stand: 24.06.18

[St99] Stoupel E., Journal of Clinical and Basic Cardiology, 1999, https://www.kup.at/kup/pdf/26.pdf, Stand: 24.06.18

Neuronale Netze

Automatic Aortic Wall Segmentation and Plaque Detection using Deep Convolutional Neural Networks

Marcel Beetz[1]

Abstract: Abnormal aortic wall thickness and the presence of aortic plaque have been linked to various types of cardiovascular disease. Quantification of both indicators currently depends on manual or semi-automatic methods which suffer from limited quality and long acquisition times. This work presents various fully automatic state-of-the art solutions to two medical image processing problems: aortic wall segmentation and plaque slice detection. A u-net derived residual convolutional neural network (CNN), a cascaded pipeline of two CNNs and a 3D CNN architecture are used for aortic wall segmentation. Plaque detection is performed by a standard multilayer residual CNN classification architecture, a u-net derived CNN classifier and a capsule CNN. The experiments show that the u-net inspired residual CNN performs best at the aortic wall segmentation task with a Dice score of around 0.8 while the capsule CNN achieves the best results in slice-wise plaque detection with a precision of 0.74 and an accuracy of 0.68.

Keywords: Deep Learning; Convolutional Neural Networks; Capsule Networks; Aortic Wall Segmentation; Plaque Detection; Medical Image Processing

1 Introduction

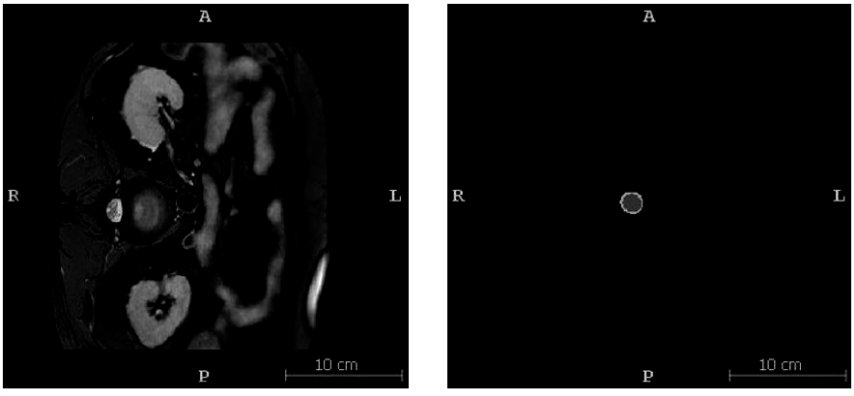
Cardiovascular disease continues to be the most common cause of death in Europe [Wi17]. Both aortic wall thickness and aortic plaque have been shown to be potential indicators in predicting and identifying cardiovascular disease [Gu10] [Ho16]. More specifically, abnormal aortic wall thickness has been linked to medical conditions such as atherosclerosis [Li15] while abdominal aortic plaque has been associated with coronary artery disease [Li16]. In both cases, there is limited automation in current medical practice with only semi-automatic methods being used to obtain segmentations or detect plaque. These semi-automatic techniques still require considerable human intervention leading to higher costs, human errors, inter- and intra-operator variability and lower reproducibility. In addition, medical images are generally more challenging to segment due to a smaller data set size, low and anisotropic image resolution, intensity inhomogeneities and the presence of imaging artifacts. This results in poor accuracy of traditional segmentation methods whose reliance on intensity or shape information alone is not able to capture the complexities of medical images, including those of the aorta region. Therefore, this paper proposes various fully

[1] Department of Informatics, Technical University of Munich, Boltzmannstr. 3 , 85748 Garching, Germany
marcel.beetz@tum.de

automatic methods derived from latest advances in deep learning to solve many of the aforementioned issues.
In summary the contributions of this paper are twofold:

- Three state-of-the-art fully automatic medical image segmentation methods, namely a u-net derived residual convolutional neural network (CNN), a cascaded CNN architecture and a 3D CNN, are applied to aortic wall segmentation.
- A capsule network as a recent innovation in general image classification is adapted for the detection of plaque in medical images and compared to previous state-of-the-art deep learning-based classifiers.

Both methods were applied and evaluated on a dataset of T2 abdominal images. Fig. 1 shows an example of a 512x512 input image of this dataset and the corresponding segmentation mask. The black area in the segmentation mask identifies the *background* region, while gray and white colored areas refer to the aorta's *blood-pool* and the *aortic wall* respectively. The aorta makes up only a small portion of the overall image but is still relatively easy to detect due to its recognizable shape. The aortic wall however only consists of a few pixels. This combined with the other previously mentioned challenges of medical data sets makes it difficult to find an accurate segmentation automatically. In plaque-containing slices, the *plaque* region is located on the inner side of the aortic wall with an average diameter of a few voxels and a partially circumferential shape.



a) Image b) Segmentation mask

Fig. 1: Example of a 512x512 input image (a) and its corresponding segmentation mask (b)

2 Previous work

Recently, deep convolutional neural networks have been shown to outperform traditional methods on a variety of data sets in both image segmentation and classification [Po11].

Common benchmarks such as MNIST for handwritten digit classification or the annual ImageNet Large Scale Visual Recognition Challenge (ILSVRC) [Ru15] to detect real-world objects from 200 categories have seen tremendous improvements achieved by deep learning-based methods. AlexNet [KSH12] was the first CNN to be applied to image classification, winning the 2012 ILSVRC by an error rate of more than 10% lower than the second best method. It introduced dropout layers and data augmentation to combat model overfitting. VGG net [SZ14] decreased the kernel size of the convolutional layers and showed the feasibility of very deep architectures. ResNet [He16] was the first model to overcome average human level accuracy with a classification error rate at ILSVRC of 3.6%. It introduced residual blocks to enable easier gradient flow and therefore network training allowing it to further increase its architectural depth. A recent advancement have been capsule CNNs which showed an equal and in some cases a better performance than current state-of-the-art methods on the MNIST dataset [SFH17]. In image segmentation, fully convolutional neural networks (FCN) [LSD15] enabled a pixel-to-pixel mapping from input image to the output segmentation mask thereby improving upon previous region-based segmentation methods. Specifically regarding medical images, the u-net [RFB15] allowed for CNNs to be used in a complete image segmentation pipeline by combining a down-sampling path with normal convolutional layers and an up-sampling path with transpose convolutions to obtain a segmentation mask with the same dimensions as the input image in one pass over the network. Further advances introduced short residual connections and more complex combinations of convolutional, normalization and activation layers [Dr16]. More recently, both cascaded segmentation pipelines [Ch17] and 3D convolutional neural networks [MNA16] have been used to improve medical image segmentation results even further. Deep CNNs have also been successfully applied in medical imaging as a classifier for various types of cancer (e.g. breast cancer [Ar17]). However, both aortic wall segmentation and plaque detection have seen limited use of deep learning-based methods and have been most commonly performed via semi-automatic approaches based on graph cuts [Du12] or geodesic active contour models [Wa17].

3 Methods

This section describes the general data preparation steps, the three approaches to each of the two main problems examined in this paper and the evaluation metrics.

3.1 Data preparation

Various widely used preprocessing steps were applied to the dataset to ease comparability of data instances among each other and improve the network's learning and prediction capabilities. First, voxel spacing was adjusted to a common value among all data instances and the few images with a different resolution than 512x512 were resampled to that value. Next, slice-wise normalization of intensity values was applied to facilitate the network's

learning process. In addition, histogram equalization was used in order to enhance image contrast. The segmentation masks associated with the raw images of wrong resolution were also resampled to 512x512 using the nearest neighbor method to maintain the whole numbers that identify each image region. Voxel spacing was adjusted for the segmentation masks as well. Data augmentation resulted in worsened performance compared to solely relying on the input data set and was therefore not used for the methods presented in the following.

3.2 Basic CNN architecture

The basic CNN architecture in this paper is derived from the popular u-net [RFB15] and work by Drozdzal et al. [Dr16] (Fig. 2 (a)). It consists of a downward contracting path to extract relevant features followed by an upward expanding path to recreate a predicted segmentation mask with the same dimensionality as the original input image. Each additional convolutional layer along the contracting path has the task of learning increasingly complex features. Long skip connections between the downward and upward path concatenate the respective feature maps to allow for better information retention between the paths. The dimensionality of the feature maps after each level is provided in Fig. 2 (a) with the first two numbers showing the feature map's x and y dimension and the third number depicting the number of feature maps used.

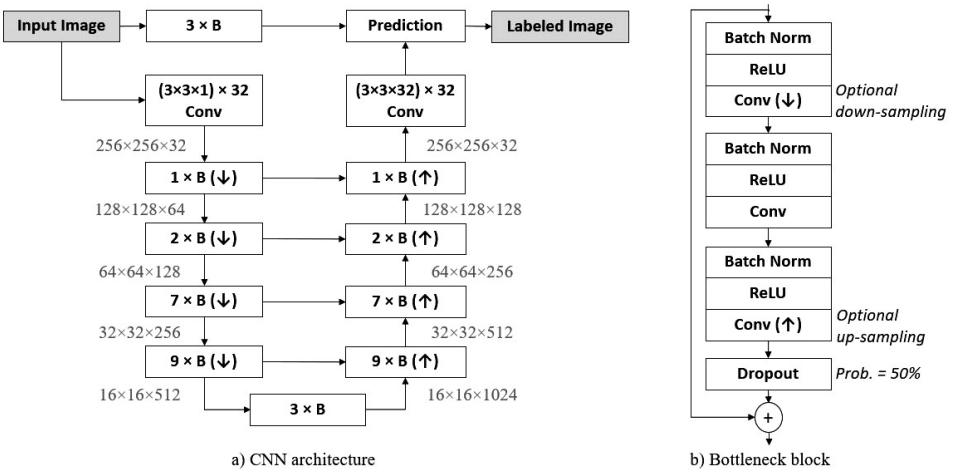


Fig. 2: Overview of complete CNN (a) and bottleneck block architecture (b)

Each layer in the CNN architecture consists of a number of bottleneck blocks which in return are made up of three iterations of a batch normalization, an activation and a convolutional layer followed by a dropout layer at the end (Fig. 2 (b)). Special convolutional layers with a stride value of 2 are used for down- or up-sampling the feature map resolution, which is indicated in Fig. 2 (b) by an up- or downward pointing arrow respectively. In addition, a

short residual connection is applied around the entire block to also allow information flow at a smaller scale (Fig. 2 (b)). Due to memory limitations, a 256x256 sized fixed ROI around the aorta is cropped out of the original image slices and used as an input to the CNN.

3.3 Cascaded CNN pipeline

Cascaded CNN pipelines have been successfully applied to various tasks in medical imaging [Ch17]. In this work a first CNN is trained to localize the aorta within the whole T2 abdominal scan (Fig. 3, left). This is necessary because there is a considerable variation between aorta positions of different slices. After the position has been found, the aorta region is cropped out of the whole input image and resized to 64x64. The resulting slices now mostly depict the actual aorta with little surrounding tissue, allowing the network to focus on segmenting the clinically important aorta region.
Next, a second CNN is used to find the final segmentation mask using the smaller cropped out aorta images of the first CNN as an input (Fig. 3, right). This specialization allows each of the two networks to focus on learning the appropriate weights for its specific task.

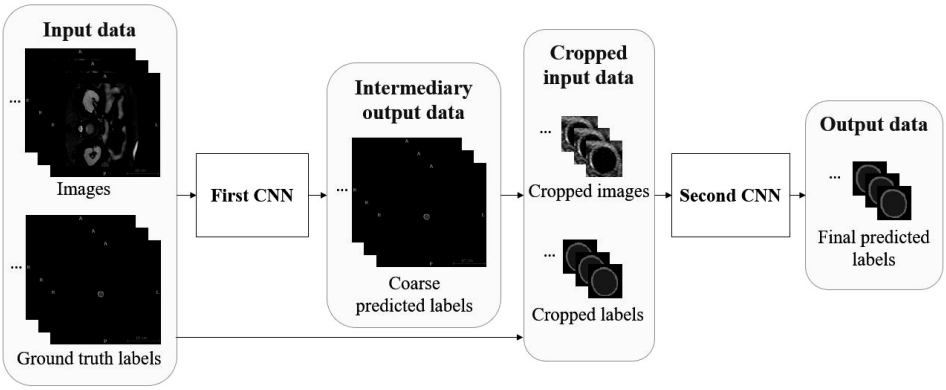


Fig. 3: Cascaded CNN pipeline

3.4 3D CNN

Similar to the previously depicted methods, 3D CNNs have also shown promising results in medical image segmentation [Ha17] [MNA16]. The access to additional information in the third dimension allows 3D CNNs to learn better, more discriminative features. The architecture used in this work is based on the 2D cascaded CNN pipeline of the last chapter (Fig. 3) with all 2D convolutional layers of the second CNN replaced by 3D ones. Also, in order to perform the memory-intensive 3D convolutions, the architecture was simplified by removing one level at the top of the architecture, resulting in a smaller overall depth of 4 levels, and by reducing the number of bottleneck blocks and channels in the other levels.

3.5 CNNs for slice classification

The second problem tackled in this work is to classify whole slices as containing plaque or not. Given the u-net inspired architecture of 3.2 a straightforward way of achieving a whole slice classification is to replace the prediction block with a fully connected layer followed by a softmax layer to get prediction probabilities for each of the two classes. This approach was the first method used in this work. As in the 3D architecture and all following classification methods, it is applied to the cropped out 64x64 aorta images and therefore acts as a second CNN in the cascaded pipeline. However, in other literature the upward path of the CNN is usually discarded for slice classification resulting in the second architecture used for plaque detection, which is depicted in Fig. 4.

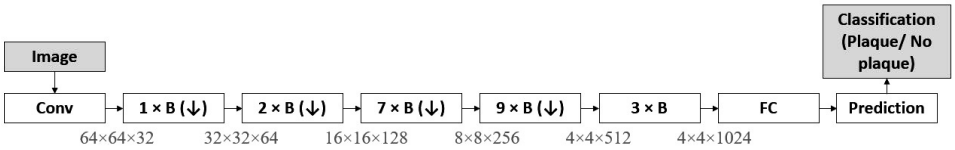


Fig. 4: CNN plaque slice classification architecture

Hereby, only the former downward contracting path is used and a fully connected layer followed by a softmax prediction layer are added to the end to obtain the class probabilities.

3.6 Capsule CNN

As a third classification method this paper adapts and applies the promising capsule networks to the more challenging medical imaging data. Similar to the original paper [SFH17], the architecture is less complex than comparable state-of-the-art CNNs and consists of two convolutional layers followed by a fully connected layer. The first convolutional layer has 256 filters with 9x9 kernels, stride 1 and is followed by a ReLU activation function. The second convolutional layer consists of 32 8-dimensional capsules and the final layer has a 16-dimensional capsule for each of the two output classes.

3.7 Quality measures

The quality of each presented segmentation method is evaluated via the Dice score metric [Di45] which measures the overlap of the predicted region with the ground truth region. It is defined in (1), where T is the ground truth region and P the predicted region:

$$dice = \frac{2 \times |P \cap T|}{|P| + |T|} \tag{1}$$

The Dice score assumes values between 0 and 1 with 0 equating to the worst and 1 to the best possible segmentation. The quality of the classification methods is analyzed using the metrics precision, accuracy and recall (2, 3, 4) [Po11].

$$precision = \frac{tp}{tp + fp} \tag{2}$$

$$accuracy = \frac{tp + tn}{tp + tn + fp + fn} \tag{3}$$

$$recall = \frac{tp}{tp + fn} \tag{4}$$

All three metrics are derived from the confusion matrix which gives the number of true positive (tp), true negative (np), false positive (fp) and false negative (fn) slices when comparing ground truth with predicted classification of plaque-containing slices.

4 Results

This section provides a short overview of the dataset, the setup of the experiments and the results of all 3 segmentation and all 3 slice-wise classification methods.

4.1 Dataset

The dataset consisted of 240 abdominal T2 scans acquired over a duration of several years. Each scan was composed of slices with the dimensionality of 512x512. A label mask associated with each scan that identified four different image regions, namely *background*, *blood-pool*, *aortic wall* and *plaque* was used as the ground truth. The dataset was randomly split into 50% training and 50% testing data to allow evaluation of the proposed methods. Hereby, the class imbalance between plaque-containing slices and non-plaque-containing slices in the overall dataset was maintained after the train-test split to retain realistic training and test conditions and enable results that generalize well.

4.2 Experimental setup

All experiments were run on a 12 GB Nvidia GPU with the tensorflow deep learning library. Adam was used as an optimizer for training. Experiments were run with three different loss functions, i.e. Dice loss, Jaccard loss and cross entropy loss. Cross entropy performed best

in all instances and was therefore used as a loss function in all reported results. The learning rate was set to 0.0005 and the batch size to 4 in all cases except the 3D model where it was set to 2 due to memory restrictions. Both values represent the optimum found after various experiments. The three aortic wall segmentation methods were evaluated via the standard Dice coefficient and the three classification methods via the metrics precision, accuracy and recall.

4.3 Aortic wall segmentation

Overall, all deep residual CNN architectures achieved good results in the aortic wall segmentation task. An example qualitative segmentation result is shown in Fig. 5. Both location and curvature of the aortic wall are captured very well by all CNN methods. Shortcomings can be seen at some portions of the aortic wall where the wall thickness of the network prediction appears slightly larger than in the corresponding ground truth image.

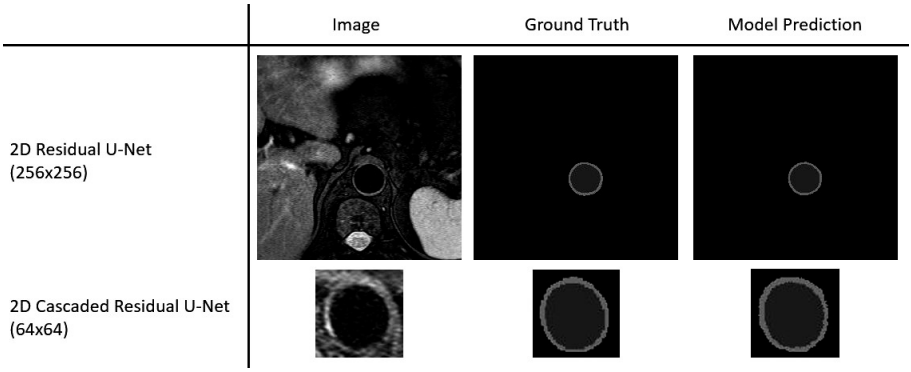


Fig. 5: Example of qualitative segmentation results

Quantitative results are presented in Tab. 1 with the u-net derived architecture of 3.2 in the first row. The next row refers to the cascaded architecture of 3.3 with a 2D CNN used as a second network while the last row corresponds to the cascaded CNN architecture with a 3D CNN used as a second CNN. Each of the three columns represents one of the aforementioned image regions.

Segmentation method	Background	Blood-pool	Aortic wall
Basic CNN	**0.999**	**0.958**	**0.797**
Cascaded 2D	0.991	**0.958**	0.791
Cascaded 3D	0.989	0.954	0.768

Tab. 1: Dice scores of different segmentation methods for various image regions

The *blood-pool* region was segmented very well by all examined methods with the basic CNN and the cascaded 2D method performing slightly better than the cascaded 3D technique.

The clinically most relevant region *aortic wall* was also segmented well by all methods with the basic CNN again achieving the highest score, closely followed by the cascaded 2D method.
Therefore the overall best performing method for aorta segmentation in this work was the basic CNN architecture, which achieved the highest Dice scores for all regions in question.

4.4 Plaque detection

Three different slice-wise plaque detection methods were analyzed using the precision, accuracy and recall metrics (Tab. 2). The first line in the table refers to the residual CNN without an upward path of Fig. 4. The second line depicts the results of the residual slice-wise classification u-net (3.5) while the third line indicates the capsule CNNs performance (3.6).

Classification Method	Precision	Accuracy	Recall
Residual CNN	**0.75**	0.61	0.25
Residual U-Net	0.66	0.67	0.27
Capsule CNN	0.74	**0.68**	**0.29**

Tab. 2: Precision, accuracy and recall of each of the three plaque slice classification methods

All methods achieved considerably better performance than random guessing. The residual CNN had the highest precision value with the capsule CNN only slightly behind. In both accuracy and recall the capsule CNN achieved the highest scores closely followed by the residual u-net in both cases. All in all, the best method for the binary classification of plaque slices in this work was therefore the capsule CNN, which had an almost as high precision score as the residual CNN and both the highest accuracy and recall scores.

5 Discussion

This section provides a discussion and potential explanations of the observed results.

5.1 Aortic wall segmentation

The basic CNN architecture performed best in segmenting all three different image regions. One major advantage of this method was that no resizing had to be performed to adjust the differently sized cropped aorta images to one resolution that the CNN could accept as input, which had to be done in both cascaded methods. This allowed the data to retain the highest degree of realness and not lose important textural information due to the inherent errors of resizing. Due to the aforementioned general complications of medical imaging data, this avoidance of further image degradation turned out to be decisive in the segmentation

task. Also, the larger 256x256 static ROI crop in the basic CNN allowed for a bigger area of the aorta's naturally surrounding tissue to remain within the image, which might have helped the CNN in better delineating the wall boundaries. This is in contrast to the original assumption that by cropping out as much surrounding tissue around the aorta as possible, the network would be able to focus on segmenting the important aorta region itself and not get distracted by other unimportant tissue. While this assumption might still hold, its gain was not enough to offset the errors induced by resizing.
In addition, considering multiple slices when performing a segmentation as in the 3D cascaded method worsened performance, indicating that there was a high variation in aortic wall characteristics between neighboring slices which confused the CNN instead of helping it learn from the additional data points of adjacent slices.

5.2 Plaque detection

The plaque detection task was best achieved by the capsule CNN. As compared to standard CNNs, capsule CNNs are better able to handle viewpoint changes of objects, i.e. to identify an object as the same object type even if the object itself was rotated or translated. Since plaque changes considerably in size, shape and position between individual slices, this property allows the capsule CNN to better detect its existence due to it storing a more complete representation of what constitutes plaque in the form of vectors as opposed to scalar values in the case of a standard CNN. Another advantage of capsule CNNs was their simpler architecture consisting of only three different layers, indicating a more efficient design and potentially leading to shorter training and running times. The residual CNN with no upward path performed considerably better on precision than the residual u-net because no information was lost in the transpose convolution operations of the upward path. While these operations are necessary in case of a full slice segmentation to obtain a mask with equal resolution to the input image, this work showed that they are detrimental when performing whole-slice classification.

6 Conclusion

This work examined various state-of-the-art methods for two problems in medical imaging. A u-net derived residual CNN with an increased number of layers and an organisation in bottleneck blocks achieved the best results in segmenting background, bloodpool and aortic wall in abdominal T2 images. Cascaded 2D and 3D approaches performed worse due to image quality loss in resizing and lack of surrounding tissue information due to narrower cropping around the aorta. The recently published capsule CNNs performed better than previous state-of-the-art methods in binary slice classification of challenging medical images due to their capacity to better represent and detect changes in plaque size, shape and position.
In the future, the promising but relatively simplistic capsule CNN architecture could achieve

even better results with a more elaborate and fine-tuned architecture design. Capsule CNNs could also be extended to perform full image segmentation instead of just classification. Furthermore, other recent advances in deep learning, such as atrous convolutions, might capture differences in shape, size and location in a better way due to their less static and more flexible kernels.

References

[Ar17] Araújo, T.; Aresta, G.; Castro, E.; Rouco, J.; Aguiar, P.; Eloy, C.; Polónia, A.; Campilho, A.: Classification of breast cancer histology images using Convolutional Neural Networks. PloS one 12/6, e0177544, 2017.

[Ch17] Christ, P. F.; Ettlinger, F.; Grün, F.; Elshaera, M. E. A.; Lipkova, J.; Schlecht, S.; Ahmaddy, F.; Tatavarty, S.; Bickel, M.; Bilic, P., et al.: Automatic liver and tumor segmentation of ct and mri volumes using cascaded fully convolutional neural networks. arXiv preprint arXiv:1702.05970/, 2017.

[Di45] Dice, L. R.: Measures of the amount of ecologic association between species. Ecology 26/3, pp. 297–302, 1945.

[Dr16] Drozdzal, M.; Vorontsov, E.; Chartrand, G.; Kadoury, S.; Pal, C.: The importance of skip connections in biomedical image segmentation. In: Deep Learning and Data Labeling for Medical Applications. Springer, pp. 179–187, 2016.

[Du12] Duquette, A. A.; Jodoin, P.-M.; Bouchot, O.; Lalande, A.: 3D segmentation of abdominal aorta from CT-scan and MR images. Computerized Medical Imaging and Graphics 36/4, pp. 294–303, 2012.

[Gu10] Gupta, S.; Berry, J. D.; Ayers, C. R.; Peshock, R. M.; Khera, A.; De Lemos, J. A.; Patel, P. C.; Markham, D. W.; Drazner, M. H.: Left ventricular hypertrophy, aortic wall thickness, and lifetime predicted risk of cardiovascular disease: the Dallas Heart Study. JACC: Cardiovascular Imaging 3/6, pp. 605–613, 2010.

[Ha17] Havaei, M.; Davy, A.; Warde-Farley, D.; Biard, A.; Courville, A.; Bengio, Y.; Pal, C.; Jodoin, P.-M.; Larochelle, H.: Brain tumor segmentation with deep neural networks. Medical image analysis 35/, pp. 18–31, 2017.

[He16] He, K.; Zhang, X.; Ren, S.; Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Pp. 770–778, 2016.

[Ho16] Hoffmann, U.; Massaro, J. M.; D'Agostino, R. B.; Kathiresan, S.; Fox, C. S.; O'Donnell, C. J.: Cardiovascular event prediction and risk reclassification by coronary, aortic, and valvular calcification in the Framingham Heart Study. Journal of the American Heart Association 5/2, e003144, 2016.

[KSH12] Krizhevsky, A.; Sutskever, I.; Hinton, G. E.: Imagenet classification with deep convolutional neural networks. In: Advances in neural information processing systems. Pp. 1097–1105, 2012.

[Li15] Liu, C.-Y.; Chen, D.; Bluemke, D. A.; Wu, C. O.; Teixido-Tura, G.; Chugh, A.; Vasu, S.; Lima, J. A.; Hundley, W. G.: Evolution of aortic wall thickness and stiffness with atherosclerosis: long-term follow up from the multi-ethnic study of atherosclerosis. Hypertension/, HYPERTENSIONAHA–114, 2015.

[Li16] Li, W.; Luo, S.; Luo, J.; Liu, Y.; Huang, W.; Chen, J.: Association between abdominal aortic plaque and coronary artery disease. Clinical interventions in aging 11/, p. 683, 2016.

[LSD15] Long, J.; Shelhamer, E.; Darrell, T.: Fully convolutional networks for semantic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. Pp. 3431–3440, 2015.

[MNA16] Milletari, F.; Navab, N.; Ahmadi, S.-A.: V-net: Fully convolutional neural networks for volumetric medical image segmentation. In: 3D Vision (3DV), 2016 Fourth International Conference on. IEEE, pp. 565–571, 2016.

[Po11] Powers, D. M.: Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation./, 2011.

[RFB15] Ronneberger, O.; Fischer, P.; Brox, T.: U-net: Convolutional networks for biomedical image segmentation. In: International Conference on Medical image computing and computer-assisted intervention. Springer, pp. 234–241, 2015.

[Ru15] Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; Fei-Fei, L.: ImageNet Large Scale Visual Recognition Challenge. International Journal of Computer Vision (IJCV) 115/3, pp. 211–252, 2015.

[SFH17] Sabour, S.; Frosst, N.; Hinton, G. E.: Dynamic routing between capsules. In: Advances in Neural Information Processing Systems. Pp. 3859–3869, 2017.

[SZ14] Simonyan, K.; Zisserman, A.: Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556/, 2014.

[Wa17] Wang, Y.; Seguro, F.; Kao, E.; Zhang, Y.; Faraji, F.; Zhu, C.; Haraldsson, H.; Hope, M.; Saloner, D.; Liu, J.: Segmentation of lumen and outer wall of abdominal aortic aneurysms from 3D black-blood MRI with a registration based geodesic active contour model. Medical image analysis 40/, pp. 1–10, 2017.

[Wi17] Wilkins, E.; Wilson, L.; Wickramasinghe, K.; Bhatnagar, P.; Leal, J.; Luengo-Fernandez, R.; Burns, R.; Rayner, M.; Townsend, N.: European cardiovascular disease statistics 2017. European Heart Network: Brussels, Belgium/, 2017.

Hyper-Parameter Search for Convolutional Neural Networks – An Evolutionary Approach

Victoria Bibaeva[1]

Abstract:

Convolutional neural networks is one of the most popular neural network classes within the deep learning research area. Due to their specific architecture they are widely used to solve such challenging tasks as image and speech recognition, video analysis etc. The architecture itself is defined by a number of (hyper-)parameters that have major impact on the recognition rate. Although much significant progress has been made to improve the performance of convolutional networks, the typical hyper-parameter search is done manually, taking therefore a long time and likely to disregard some very good values. This paper solves the problem by proposing two different evolutionary algorithms for automated hyper-parameter search in convolutional architectures. It will be shown that in case of image recognition these algorithms are capable of finding architectures with nearly state of the art performance automatically, sparing the scientists from much tedious effort.

Keywords: deep learning; convolutional neural networks; CNN; hyper-parameter search; evolutionary algorithms; genetic algorithm; memetic algorithm

1 Introduction

Recent decade brought an enormous scientific progress in the field of deep learning, which deals with deep, many-layered neural networks and their learning techniques. One of the most prevalent classes of such networks is Convolutional Neural Networks (**CNNs**). A CNN is a specific type of multi-layer perceptron (**MLP**) with more complex architecture. It was first proposed in 1989 as a model inspired by visual cortex of mammals, and was used to classify the images of hand-written digits [Le98]. In 2012 came a great breakthrough, as a CNN suggested by [KSH12] won the world's biggest object recognition challenge ILSVRC. It successfully classified natural objects in a dataset containing of 1.4 million images and 1000 categories. Since then, CNNs quickly became state of the art and were applied to such tasks as image and speech recognition, object classification and video analysis. Nowadays they are ubiquitous due to their outstanding performance and well-developed techniques of its improvement, in some cases already reaching the error rate of humans [KSH12]. However, the growing complexity of CNN architectures causes many problems, an important

[1] Hochschule für Angewandte Wissenschaften Hamburg (HAW Hamburg), Department Informatik, Berliner Tor 5, 20099 Hamburg, Germany firstname.lastname@haw-hamburg.de

one being "*overfitting*". It happens when the network memorizes the dataset, but is not able to recognize slightly different objects outside of the dataset.

The performance of CNNs on an object classification task can be defined as recognition rate, i.e. percentage of correctly classified images. It can be influenced through the choice of dataset, but also through changing the training parameters (like learning rate of gradient descent) or carefully tuning the network architecture (represented by all the corresponding **hyper-parameters**). Whereas the first two influence factors are generally well-studied [Hi12], the architecture tuning is typically carried out manually in a haphazard manner, considering merely a small number of proven values and techniques. Only few scientific studies are dedicated to the empirical impact of hyper-parameters (cf. [Ja09, MSM16]), while their mutual influence remains largely unknown. Thus, there is still a lack of knowledge concerning reasons behind a good performance of CNNs and its further improvements.

A sensible approach to design a CNN to solve any given task is to try out not only one, but many different hyper-parameter combinations. This is likely to result in a model with higher recognition rate. Yet the manual search for good hyper-parameter values can be very time and resource consuming, not to mention the chance of overseeing some promising values. An automated hyper-parameter search, on the other hand, has a potential to save a lot of tedious effort using some intelligent search strategy. Our work originated from an idea to implement such an algorithm using the up-to-date knowledge about CNNs. Two algorithms presented in this paper are based on evolutionary computation and have already shown their worth in finding good architectures of MLP. It will be demonstrated that they can be successfully adjusted to the case of CNNs, providing better CNN architectures than a baseline method and achieving nearly state of the art performance without any user intervention. By means of some representative datasets from the image recognition domain we will determine, which algorithm gives better results under certain conditions. Such criteria as architecture quality, complexity and runtime will be evaluated as well.

Our paper is organized as follows. First, we will introduce different hyper-parameters of a CNN and their possible values. Then, in section 3, the current methods of hyper-parameter search will be reported in case of CNNs as well as MLP. Section 4 gives detailed account on the two proposed evolutional algorithms. At last, the major results of our experiments are presented in section 5, leaving the last section for conclusion and future work.

2 Convolutional Neural Networks

In order to classify objects a CNN receives an input image and extracts certain visual features from it. The extraction is done step by step, so that the features extracted during the succeeding layers (rectangles, circles) are composed of features from the preceding layers (lines, angles, arcs). Thus, a feature hierarchy is gradually constructed [Le98]. Therefore a CNN contains of several so-called **feature extraction stages**, each of which is composed of three basic types of neuron layers [Ja09]: *filter bank layer*, *non-linearity layer*, *feature*

pooling layer. After the feature extraction stages follows a **classifier**, which is essentially a MLP with some *full connection layers* that calculates the probability of the object class. A famous instance of a CNN architecture is shown in Fig. 1. It consists of two feature extraction stages (C1 + S2, C3 + S4) and a three-layered classifier (C5 + F6 + Output).

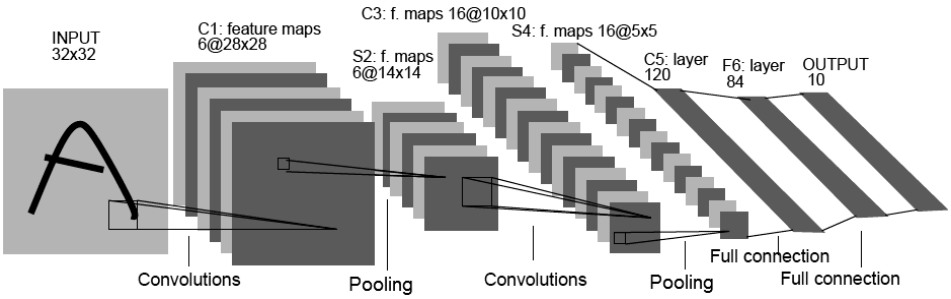


Fig. 1: CNN architecture [Le98].

So how can we generate variations of any given CNN architecture? In other words, what are the tunable hyper-parameters? These are, on the one hand, the number of feature extraction stages and their specific components (layer types and their sequence), but also the number of hidden layers in the classifier. The rest of this section will be dedicated to the description of different layer types and the corresponding hyper-parameters.

Filter Bank Layer This layer type is where the convolution operation takes place, to which CNNs owe their name [Ja09]. Convolution produces a mapping between a source image and target image by applying a convolutional kernel (also named "*filter*") onto overlapping regions of the source image. Thus, a filter highlights one certain feature in the entire image. Unlike MLP, the neurons within one filter bank layer are divided into several planes. All neurons in one plane share weights and perform the same convolution operation in order to extract one certain image feature. Classifying more complex objects requires more features and therefore more filters in one layer – hence the term "filter bank". Filter bank layer has 3 major hyper-parameters, namely: filter size, step size (to define the overlapping image regions) and filter count. Fortunately, not many values of these hyper-parameters have prevailed in common practice. Nevertheless, there are only rules of thumb to set the number of filters, depending on the expected number of features and object complexity [MSM16].

Non-linearity Layer Traditionally, the output of filter bank layer is then fed to non-linearity layer, which is often considered to be a part of the former [Ja09]. As in a common neuron, a non-linear activation function is applied here to each pixel value, in order to propagate or suppress the incoming visual signal. A very popular function is $g(x) = max(0, x)$, and the neurons using it are named ReLU ("rectified linear units", [KSH12, Hi12]). Nonetheless, a variety of other functions based on ReLU were presented in recent years. The study [MSM16] compared these functions within one particular CNN architecture being trained

on ILSVRC dataset. The lowest error rates were achieved by ELU ("exponential linear unit"), maxout and their combination (see [MSM16] for references). Accordingly, we have chosen these 4 functions for our trials, adding one hyper-parameter to a current layer type.

Feature Pooling Layer The last layer type of a feature extraction stage reduces the source image resolution even further, making the outcome independent of the exact feature position. It is done by applying the same filter onto the overlapping regions of the image. In contrast to filter bank layer, this filter is not learned during training, but calculates maximal or average pixel value of every region [Hi12]. Therefore, the layer is called max or average pooling layer, accordingly. The resulting target images have lower dimensions, and so a part of visual information is lost (proportionally to filter size). The choice of an appropriate pooling type (max or average) is controversial and cannot be answered in general for all the CNN architectures or datasets. Furthermore, there is evidence, that the sum of max and average pooling can be even more effective [MSM16]. As a result we consider 3 hyper-parameters of a pooling layer: filter size, step size and pooling type with 3 possible values.

Full Connection Layer As stated earlier, the classifier of a CNN aims to learn the relationship between the extracted features and the resulting object classes. It contains of several stacked full connection layers without weight sharing. This implies the biggest number of trainable weights in the whole network and the necessity to limit the count of such layers, typically using only 2 or 3 (cf. [KSH12, Ja09, MSM16]). Each full connection layer may be followed by a non-linearity layer, leading us to define 2 hyper-parameters: number of output neurons and type of activation function. We also used Dropout with probability 0.5 (see [Hi12]) after each full connection layer as a technique to avoid overfitting.

Hyper-Parameter Overview The correct order of layer types within one feature extraction stage has a very important role in designing a CNN. It was already studied in [Ja09], where the lowest error was achieved by CNNs with layer sequence "Filter Bank → Non-Linearity → Feature Pooling". Unsurprisingly, this layer sequence actually occurs in nature. All hyper-parameters introduced above and their possible values are summed up in Tab. 1. However, the interaction between the hyper-parameters remains largely unknown. The existing reports on this subject (eg. [Ja09, MSM16]) used a single dataset for evaluation and noted that their results cannot be automatically transferred to other datasets.

3 Related Work

The goal of object classification is to design a model, in our case a CNN, which after training *generalizes* the given dataset well. It means, the model can correctly classify images that were not included in the training set. For this purpose, a test set is employed, which is also a

Hyper-Parameters						Ref.
Filter Bank Layer	Activation Function (ReLU, ELU, Maxout, ELU + Maxout)	–	Filter Size (3, 5, 7, 9, 11)	Step Size (1, 2, 3, 4, 5)	No. of outputs	[Hi12]
Feature Pooling Layer	–	Pooling (Max, Average, Max + Average)	Filter Size (2, 3)	Step Size (1, 2)	–	[KSH12]
Full Connection Layer	Activation Function (ReLU, ELU, Maxout, ELU + Maxout)	–	–	–	No. of outputs	

Tab. 1: Hyper-parameters used in this paper and their values.

part of the given dataset, but is not generally used for training. So the quality of our model can be summed up as **accuracy** on the test set, and we are aiming to find a model with the highest accuracy (percentage of correct classifications). On the other hand, the objective function of gradient descent during training is the so-called **loss** function which measures error on training set. Both loss and accuracy indicate how well the given model performs, accuracy being more important criterion for the future productive use of the model [Le98]. Other quality criteria such as training time, computational complexity or memory usage are of less importance, because they depend on the chosen network implementation tool.

Previous section illustrated the fact that there is an enormous number of different CNNs applicable for any given dataset. It makes training and testing them all quite impossible, considering that each training cycle might take several days or weeks. Even on the modern hardware this task can be insurmountable, leading to the necessity of using an automated hyper-parameter search algorithm. Its benefits would be sparing time and resources due to an intelligent search strategy and minimizing the risk of overseeing promising hyper-parameter values. On the other hand, it should be able to leverage the existing knowledge about CNNs to avoid certain pitfalls and produce models with above-average quality. The real challenge is to find an algorithm which quickly and reliably finds a competitive architecture.

Surprisingly, relatively little scientific research is done to solve this problem yet, see [Be11, BB12, Sn15]. These studies were motivated by the fact that the advances in object classification can be achieved through hyper-parameter tuning, rather than inventing new models or training techniques. In the authors' opinion, the hyper-parameter search is nowadays "more of an art than a science", even though it should belong to the designing process of each deep learning model [Be11]. Thus, they proposed Grid Search and Random Search as an alternative to manual hyper-parameter exploration.

If the desired values of hyper-parameters are known in advance, then all their conceivable combinations can be easily generated, essentially defining all the points in the initial search

space. **Grid Search** [BB12] reduces the overall search space by selecting a few values of each hyper-parameter. If combined with each other, these selected values create a grid within the search space. Subsequently, grid points are considered as solution candidates, and the corresponding architectures are tested to identify the model with the best accuracy. Advantages of Grid Search are simple implementation and a possibility of parallel processing.

As opposed to Grid Search, **Random Search** selects some random combinations of hyper-parameters. These are basically random points in the search space which may get closer to the better solutions than grid points. The experiments in [BB12] confirm that Random Search requires less candidates than its counterpart in order to reach certain accuracy level. However, the probability to randomly encounter a very good solution is inverse proportionate to spatial dimensions. Furthermore, Random Search is not capable of pursuing any promising directions or selectively explore certain areas. Hence, both Random and Grid Search contain an inherent tendency to step over the best solutions.

The problem of finding the best hyper-parameters is far from being solved [BB12]. In spite of the availability of such methods as Bayesian Optimization [Sn15], Grid and Random Search remain the tools of choice. Thus, there is still a need to exploit new algorithms from other domains, including less complex networks like MLP, where hyper-parameter optimization has been in research focus for a long time. The most widely used methods to find the best MLP architectures are evolutionary algorithms [CG11, OI11]. Consequently, our approach to solve the hyper-parameter search problem is based on evolutionary algorithms and the assumption that they can be adjusted to the case of CNN architectures. Two such algorithms will be presented in the following section: both are well-studied, straightforward to implement and provide an efficient search strategy that avoids local optima.

4 Proposed Algorithms

4.1 Genetic Algorithm

Genetic Algorithm (**GA**) was introduced in the 1970s [Ch11] and is since then the most well-known evolutionary algorithm. It is population-based and attempts to replicate the processes of natural evolution. Accordingly, the population individuals with higher fitness, i.e. ability to adapt to the environment, have more chances to pass their best characteristics to the next generation, whereas the unfavorable characteristics rather disappear [CG11].

A lot of articles have been published concerning hyper-parameter search for MLP using GA [OI11]. Generally, GA works as follows. Firstly, MLP architectures should be represented with binary strings in order to serve as individuals in the population. Each hyper-parameter value is expressed through one or more genes (0 or 1), so the MLP architecture can be transformed into a *chromosome*. Every iteration of GA changes the current population of chromosomes/individuals by applying genetic operators to them: selection, crossover and mutation. To measure the quality of individuals a fitness function is used, which in case of

MLP architectures can depend on training loss, test accuracy or network size [OI11]. The fitness function is subjected to maximization during the GA iterations.

Adjusting GA to the case of CNN architectures, we have chosen the following version of it:

1. Create the initial population with M random individuals
2. Evaluate the fitness of each individual
3. Apply genetic operators to the current population:
   a) Selection: The fraction p_c of the fittest individuals survive to the next generation, the rest is discarded
   b) Crossover: random pair of individuals produces one offspring by swapping some genes, until the population size is reached
   c) Mutation: p_m random genes of some individuals will be altered
4. Repeat steps 2 – 3 until the required iteration count N is reached

Genetic operators should be chosen to accomplish an efficient search strategy. Thus, **selection** attempts to enhance the average quality of the population and lead the search in the direction of promising solutions. The chosen variant of selection is based on elitism [CG11], which means that the fittest individuals should survive at any rate, not just by a given probability or ranking. It guarantees a non-decreasing fitness and genetic diversity of the population.

Crossover, on the other hand, brings variation into the population by combining good properties of selected individuals and facilitates faster convergence. The variant of crossover we used is called 1-point-crossover, as it cuts the parent chromosomes in a random position and swaps the resulting parts between the parents to produce a child chromosome [CG11]. It ensures mostly feasible and good offspring architectures, which accelerates the search and does not tear the specific CNN architecture layers completely out of their context.

The basic idea of **mutation** is to avoid local optima, intensify the population diversity and acquire new genetic material. Mutation is also affected through elitism, as the fittest individuals (selected parent chromosomes) should be excluded from changing genes [CG11].

The challenges of GA are the choice of suitable genetic operators, fitness function and binary representation of CNN architecture. Also, GA obviously contains its own specific parameters such as probability of selection and mutation, as well as population size and number of iterations. Nevertheless, the advantages of GA make it a popular tool to explore the architecture search space [Ya99]. Firstly, GA is capable to search globally, avoid local optima and guarantee the sequential fitness improvement. Secondly, it can generate architectures with the desired characteristics, by means of including them into the fitness function. In the end, GA is less sensitive with respect to initialization than one-solution-based algorithms.

4.2 Memetic Algorithm

For some instances of search problems, the GA's efficiency was reported to be unsatisfactory [KS05]. The reason is that GA lacks mechanisms to perform fine-grained search in a region with very good solutions. As opposed to global search of GA, an algorithm called **Local Search** is designed to explore the region of its initial solution in order to find the better ones. Therefore, an obvious attempts were made to hybridize GA with some form of Local Search [Ch11]. One of the most successful hybrid algorithms is Memetic Algorithm (**MA**). Its name is based on the notion of "meme", which represents a unit of cultural evolution that can exhibit local refinement [KS05]. Thus, MA uses all the key components of GA, inserting Local Search step before applying genetic operators to the current population [Ch11]. Thereby the resulting population strictly consists of local optima.

Local Search within MA is usually carried out on a subset of the population. Some neighbors of every individual in this subset are then evaluated with fitness function. If one particular neighbor has higher fitness value than the original individual, then it replaces the latter in the population. Moreover, Local Search can be done in 2 different ways [Ch11]. Firstly, the search for the individual's replacement can be continued until the first fitter neighbor is encountered. Secondly, the search can iterate over a fixed number of random neighbors, so that the complexity of one MA iteration increases as a square function of population size.

Despite higher computational complexity MA has also a number of advantages, which can be beneficial for many real-life search problem instances [KS05]. On the one hand, it has only two more specific parameters than GA, namely the number of neighbors to evaluate and the radius of neighborhood. On the other hand, as the population consists of higher quality individuals, Local Search leads to quicker convergence of MA.

In order to fully exploit the benefits of hybridization we have chosen the second, greedy variant of Local Search for our experiments. Also, we executed Local Search on the whole population, not only on its subset. These choices are consistent with the earlier established elitism, because the fittest individuals can only be replaced with even fitter ones.

5 Evaluation

We implemented the proposed algorithms in Python, using open source framework `caffe` [2] for training of CNNs. Next, a series of experiments was conducted with intention to find the best configuration for each algorithm and to assess their solution quality. For this purpose we utilized 2 datasets from object classification domain, keeping reasonable training time in mind. These datasets were chosen to represent real-world classification problems and exemplify a large number of documented results. The first dataset named CIFAR-10 is composed of 60,000 colored images of natural objects scaled to 32×32 pixel. Its 10 classes

[2] `http://caffe.berkeleyvision.org`

are: airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck (see [ZF13]). The other dataset MNIST includes 70,000 images of hand-written digits in gray scale [Le98].

As the only kind of image preprocessing we used in our work is normalization and subtracting the mean image of the entire dataset, it would be of interest to know state of the art accuracy of CNN models with the same preprocessing steps. For example, the authors of [ZF13] achieved an accuracy 0.8487 on CIFAR-10 using a specific type of pooling layer. The top score for MNIST has long been 0.9905, accomplished by the CNN architecture in Fig. 1, before it was beaten by [Ja09] with the accuracy 0.9947 and unsupervised learned filters. One can draw on these examples to estimate the reasonable layer count, which in our case varies from 6 to 8. Note that the results described below were derived from MNIST experiments, the trials with CIFAR-10 being similar are omitted due to space constraints.

The choice of the fitness function is crucial for the success of GA and MA. It should reflect the quality of a CNN architecture, duly incorporating both loss and accuracy: $Q_{fit} = -\alpha \cdot Q_{loss} + \beta \cdot Q_{acc}$, where $0 \leq \alpha < \beta$ and $\alpha + \beta = 1$. We limited ourselves to several combinations of α and β and extensively tested them, varying other specific parameters of the algorithms. The best combination turned out to be $\alpha = 0.25$ and $\beta = 0.75$, proving the hypothesis about smaller significance of loss for network quality.

Next step was to determine the best configuration for GA and MA. The number of iterations ($N = 10$) and the population size ($M = 30$) were set taking not only the available hardware into account, but also the desired variety of CNN architectures. Other specific parameters like selection and mutation ratio (p_c and p_m), shared between both algorithms, were comprehensively analyzed as well. The values $p_c = 0.5$ and $p_m = 0.75$ proved to induce the best search behavior, meaning that the fittest half of the population should be chosen to reproduce, and $^3/_4$ of their children should be subjected to mutation. As a result of more population diversity, the fitness over all iterations grew in this case always stronger than in configurations with less selected individuals or less mutation.

The remaining specific parameters of MA (the number of neighbors S to evaluate and the radius of neighborhood R) were examined separately. For instance, the radius R is calculated as the longest distance from the initial individual using its coordinates in the search space. A very good value of radius turned out to be $R = 0.15$, given the fact that it allowed us to find enough valid CNN architectures in the corresponding neighborhood. Moreover, relatively large jumps in the search space are possible, which can be very helpful to avoid local optima. We also found out that higher radius leads to longer search, as the probability of finding valid individuals decreases. In contrast, lower radius helps to preserve the local nature of the search, but unfortunately reduces the variety of neighbors, preventing MA to significantly increase the current fitness. The number of neighbors was fixed to $S = 5$ as a balance between added computational complexity and range of the search.

By setting the specific parameters of GA and MA to the aforementioned values we ensured that both algorithms perform to the best capability. Subsequently, we analyzed their properties

in order to find out how the solution was found in each case. This was accomplished by inspecting the generated architectures and the shift of their fitness values during runtime.

So how do the proposed algorithms change CNN architectures in the course of N iterations? As the initial architectures are random, there is a great variety of hyper-parameter values at the start of each algorithm. Then the variety is considerably reduced by GA, as the search quickly concentrates on one region with the fittest individuals known so far. The mechanism of crossover does not allow for significant changes in the genetic material, for which mutation remains solely responsible. So the last iteration of GA produces architectures that are very similar to each other, mostly differing in the number of outputs. MA, on the other hand, makes use of Local Search, constantly inserting new fitter neighbors into the population and thus keeping the range of genetic diversity more stable.

Another question arises: Does genetic diversity bring better results? In other words, what fitness shift can be expected from both algorithms? To answer this, we tracked the currently fittest individual in each population. Fig. 2 illustrates typical instances of fitness shift for GA and MA. Due to elitism, fitness can only increase in both cases. However, because of elitism GA does not change the fittest individuals, causing fitness plateaus as seen in Fig. 2, left. MA differs from GA in this respect, as it substitutes all individuals through their fitter neighbors, thus inducing the strictly monotonic fitness growth (Fig. 2, right).

Worthy of noting is the fact that both algorithms start with at least one very good CNN architecture in the first generation, which is a great advantage of population-based search algorithms. While the best accuracy of the first generation was rarely under 0.9, the outcoming CNNs had average accuracy above 0.97. The best observed accuracy of GA was 0.9903, MA achieved 0.9902 with less layers, approaching state of the art accuracy 0.9905.

Finally, to compare the performance of GA and MA we strategically provided them with the same preconditions, i.e. same initial population containing 7-layered CNN architectures. The averaged results of this series of trials are demonstrated in Tab. 2. As a reference algorithm

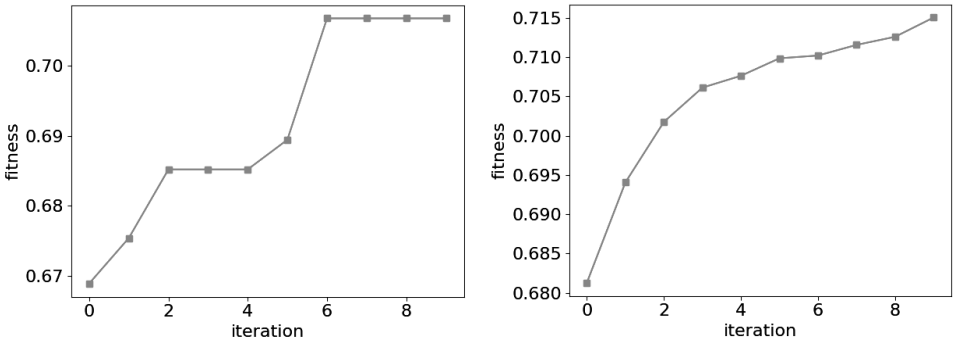


Fig. 2: The development of each generation's best fitness in GA (left) and MA (right).

to compare against we also implemented Random Search (**RS**, see section 3), starting it with the initially fittest individual of the given population and running for 100 iterations. In order to measure the success of the algorithms we have chosen a fitness increase as metrics, i.e. the difference between the best fitness of the first and the last generation. As expected, RS shows the lowest average fitness increase of all – 7.5 %. Surprisingly, GA gets only marginally better than RS (7.8 %) due to its limitations with respect to diminishing genetic diversity. The winner of these trials is MA with its ability to increase the given fitness up to 14.7 % – nearly twice as much as GA. This excellent performance can be explained with the benefits of Local Search, which enables MA to explore the neighborhood of every individual, thus avoiding too quick convergence and local optima. The average runtime of both algorithms shown in Tab. 2 should be regarded as reasonable, considering the fact that the manual search through all hyper-parameters of Tab. 1 might take longer than a week.

Algorithm	Count Specific Parameters	Mean Fitness Increase (%)	Computational Complexity	Avg. Runtime (days)	Best Known Accuracy
RS	1	7.5	linear	<1	0.9857
GA	4	7.8	quadratic	<2	0.9903
MA	6	**14.7**	cubic	5	0.9902 *

Tab. 2: Summary of our experimental results (* – achieved with less layers).

6 Conclusion and Future Work

In this paper, we analyzed the problem of finding good hyper-parameter values for CNN architectures and solved it using two evolutionary algorithms. The first one is Genetic Algorithm, the second is Memetic Algorithm, a hybrid of GA and Local Search. We demonstrated that these algorithms can be successfully adjusted to the case of CNN and even be superior to the standard search algorithms from the literature, like Random Search.

The proposed algorithms contain significantly less specific parameters than hyper-parameter amount in CNNs (4 or 6 instead of 42 in 7-layered CNNs, cf. Tab. 1, 2), making it easier to determine them manually if needed. We also illustrated the considerations behind the choice of these specific parameters. After they are set to reasonable values, both algorithms are able to efficiently explore the search space and find CNN architectures with nearly state of the art accuracy on the given dataset without any user interference. In addition, we evaluated the performance of both algorithms using fitness increase as metrics. The experiments show that GA provides stable fitness increase and improves the properties of CNN architectures with genetic operators. MA is more complex, but its performance level is twice as high as that of GA due to Local Search. Both GA and MA are population-based, independent of initialization and capable of searching for good architectures in several promising areas.

However, GA and MA can be further improved by means of parallel processing. It would facilitate the experiments with extremely large datasets like ILSVRC, which would take a week to train one CNN on. Besides, other techniques to increase the accuracy level can be

taken into account, for example, other kinds of preprocessing [Hi12, ZF13] or tuning the training parameters. In the end, the proposed algorithms can be considered as a good support for scientists who desire to improve the accuracy of their CNN models by systematically searching for appropriate hyper-parameter values.

References

[BB12] Bergstra, James; Bengio, Yoshua: Random Search for Hyper-parameter Optimization. J. Mach. Learn. Res., 13(1):281–305, February 2012.

[Be11] Bergstra, James S.; Bardenet, Rémi; Bengio, Yoshua; Kégl, Balázs: Algorithms for Hyper-Parameter Optimization. In (Shawe-Taylor, J. et al., eds): Advances in Neural Information Processing Systems 24, pp. 2546–2554. 2011.

[CG11] Correa, B.A.; Gonzalez, A.M.: Evolutionary Algorithms for Selecting the Architecture of a MLP Neural Network: A Credit Scoring Case. In: Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on. pp. 725–732, Dec 2011.

[Ch11] Chang, Y.; Wang, Y.; Ricanek, K.; Chen, C.: Feature selection for improved automatic gender classification. In: 2011 IEEE Workshop on Computational Intelligence in Biometrics and Identity Management (CIBIM). pp. 29–35, April 2011.

[Hi12] Hinton, Geoffrey E. et al.: Improving neural networks by preventing co-adaptation of feature detectors. CoRR, abs/1207.0580, 2012.

[Ja09] Jarrett, K.; Kavukcuoglu, K.; Ranzato, M.; LeCun, Y.: What is the best multi-stage architecture for object recognition? In: Computer Vision, 2009 IEEE 12th International Conference on. pp. 2146–2153, Sept 2009.

[KS05] Krasnogor, N.; Smith, J.: A tutorial for competent memetic algorithms: model, taxonomy, and design issues. IEEE Transactions on Evolutionary Computation, 9(5):474–488, 2005.

[KSH12] Krizhevsky, A.; Sutskever, I.; Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks. In (Pereira, F. et al., eds): Adv. in Neural Information Processing Systems 25, pp. 1097–1105. 2012.

[Le98] LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P.: Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11):2278–2324, Nov 1998.

[MSM16] Mishkin, Dmytro; Sergievskiy, Nikolay; Matas, Jiri: Systematic evaluation of CNN advances on the ImageNet. CoRR, abs/1606.02228, 2016.

[OI11] Oong, Tatt Hee; Isa, N.A.M.: Adaptive Evolutionary Artificial Neural Networks for Pattern Classification. Neural Networks, IEEE Transactions on, 22(11):1823–1836, Nov 2011.

[Sn15] Snoek, J. et al.: Scalable Bayesian Optimization Using Deep Neural Networks. In: Proc. of the 32nd Intl. Conf. on Machine Learning - Vol. 37. ICML'15, pp. 2171–2180, 2015.

[Ya99] Yao, Xin: Evolving artificial neural networks. Proc. of the IEEE, 87(9):1423–1447, 1999.

[ZF13] Zeiler, Matthew D.; Fergus, Rob: Stochastic Pooling for Regularization of Deep Convolutional Neural Networks. CoRR, abs/1301.3557, 2013.

Development of neural network based rules for confusion set disambiguation in LanguageTool

Markus Brenneis[1]

Abstract: Confusion set disambiguation is a typical task for grammar checkers like LanguageTool. In this paper we present a neural network based approach which has low memory requirements, high precision with decent recall, and can easily be integrated into LanguageTool. Furthermore, adding support for new confusion pairs does not need any knowledge of the target language. We examine different sampling techniques and neural network architectures and compare our approaches with an existing memory-based algorithm.

Keywords: Confusion Set Disambiguation; Grammatical Error Correction; Machine Learning; Neural Networks

1 Introduction

Grammar checkers are used to detect errors which cannot be detected by a simple spell-checker, e.g., confusion of words and agreement errors. For instance, the sentence **Then lecture is great.* contains no spelling mistake, but the words "then" and "the" have been confused. We have developed rules for confusion set disambiguation based upon neural networks and integrated them into the existing grammar checker LanguageTool.

Existing approaches for grammatical error correction have several problems: If rule based correction algorithms are used, the rules must typically be written by hand, which is time-consuming and prone to errors. Moreover, the creator of the rule must be familiar with the language the rule is written for.

On the other hand, when using an approach based on machine learning, extensive knowledge of the target language is not needed. But machine learning based grammatical error correction often suffers from creating too many false positives, which is annoying for a user of a grammar checker. Furthermore, many machine learning models require several hundreds of megabyte storage space (e.g. deep neural networks) and are slow at classification time, which both limit the usefulness of such models for end users, who often do not want a grammar checker to take up much storage space or take a long time to check a text.

[1] Heinrich-Heine-Universität Düsseldorf, Institut für Informatik, Universitätsstraße 1, Germany, markus.brenneis@uni-duesseldorf.de

Therefore, we focused on creating a classifier which has high precision to minimize false alarms, low memory requirements and little evaluation time.

We will now explain what LanguageTool is and how it works, what the task of confusion set disambiguation is, and state the goals of our work.

1.1 LanguageTool

LanguageTool is a free, open-source and rule-based grammar and style checker originally developed by Naber [Na03] and written in Java. The majority of rules are manually written in either XML or Java, hence rule development is a time-consuming task which requires knowledge of the target language.

When a text is checked, LanguageTool uses its own language-specific sentence splitter, tokenizer and part-of-speech tagger to assign part-of-speech (POS) tags to every token in the input. After POS tagging, each sentence is checked against the style and grammar rules.

1.2 Confusion Set Disambiguation

A typical type of mistake which is not detectable by a spell checker are confused words. Confusion set disambiguation is the task of choosing the right word from a finite set of words (e.g. {to, too, two}). In this paper, we will focus on confusion sets with exactly two tokens t and t'. LanguageTool already supports detecting commonly confused words. Currently, there are basically two types of rules: Pattern rules written in XML or Java, which are usually created by hand; therefore, creating new rules is time-consuming and prone to errors.

As an alternative, there are 3-gram based rules in LanguageTool, which require a copy of a large 3-gram corpus (e.g. 10 trillion tokens for English, stored in a 11 GB database) which bases upon the Google n-gram corpus [Li12]. The error detection algorithm is memory-based and works as follows: Let t be a token in a confusion pair (t, t') and t_{+n} the nth token after t in the text being checked. When t is encountered in the text, the number of occurrences m of the 3-grams (t_{-2}, t_{-1}, t), (t_{-1}, t, t_{+1}), and (t, t_{+1}, t_{+2}) are counted and compared with the number of occurrences m' of the same 3-grams containing t' instead of t. If m' is x times greater than m (where a suitable x with good precision and recall is determined beforehand), t is considered incorrect.

The 3-gram based rules have the advantage that rules have not to be written manually. On the other hand, there are several disadvantages: First, the rules fail to detect errors if the exact 3-gram is not part of the corpus. For instance, the mistake in *We go *too Gimli's birthday party.* is not detected, because the 3-grams (*go to Gimli*) and (*to Gimli 's*) are not part of the corpus, although the individual tokens are.

Furthermore, the user of LanguageTool needs to download a big corpus in order to use the rules and must have a sufficiently fast hard drive and enough memory, in order not to slow down the process of text checking too much.

1.3 Goals of our Work

The main goal of our work was to create confusion set disambiguation rules using neural networks for LanguageTool which are at least as good as the existing 3-gram based rules. Storing the new rules should not require much memory, preferable less than 100 MB, and the rules should cause as few false alarms as possible to be suitable for everyday use. What is more, the rules must be able to deal with unseen contexts and may not have a negative impact on the performance of LanguageTool. Furthermore, we wanted to examine the influence of different sampling methods and model sized on the performance of the classifier.

In the following section we will introduce our neural network architectures and the training process. Afterwards we compare our classifiers and the existing memory-based 3-gram rules with regard to precision, memory usage and speed. Finally, we have a look at alternative approaches and related work.

2 Model Architecture and Training Process

We will now describe how our classifier works and how it has been trained. In particular, we introduce our data set, discuss different approaches to sampling, our input representation and the architecture of our neural network.

2.1 Data Set

Our neural network has been trained on a large, unannotated corpus which can be considered to have no or at least very few mistakes. As shown by Banko; Brill [BB01], using larger data sets can improve the performance of a classifier significantly. Furthermore, some words like second person verb forms can only seldom be found in some corpora, for example newspaper articles. Thus, a corpus with sentences randomly chosen from newspaper articles from *Project Deutscher Wortschatz* [GEQ12] and sentences from *Tatoeba* [Ho] has been created. The final corpus for English contains more than 30,000,000 words and has been divided in a training (90 %) and testing set (10 %). This unusual split ratio has been used because we think that the data set is large enough to get decent test results with only 10 % of the data, and the model has a chance to learn more using 90 % of the data.

The corpus has been tokenized using the tokenizer of LanguageTool. For each confusion pair, we trained a separate classifier on a subset of the training set which only contains those sentences which contain the tokens of the confusion set, which typically are between 1.000 and 100.000 sentences per token, depending on how common it is in the corpus.

2.2 Sampling

It is often the case that one word of a confusion set occurs several times more often in the training corpus than the other word. Considering the German confusion set {wider, wieder}, there are around 40.000 sentences containing "wieder" in the training corpus, but only 471 sentences with "wider". Our experiments discussed in section 3.3 have shown that this class imbalance leads to heavy overfitting, since the classifier is biased towards the majority class.

Sampling Technique	occurences of *wider* in training set	occurences of *wieder* in training set
None	471	44.751
Undersampling	471	471
Oversampling	44.751	44.751
Combination Over-/Undersampling	942	942

Tab. 1: Overview of sampling techniques using the confusion set {wider, wieder} as example

To overcome the issue of class imbalance, we compared three different approaches which can commonly be found in research [Ch09]: Random undersampling, random oversampling, and a combination of over- and undersampling. In the latter case, the oversampling has been limited to a factor of 2, and the majority class has been undersampled such that the class label ratio is 1. This approached seemed to be feasible because we did not want to throw away too many training samples as would be done in undersampling, but we also wanted to prevent the classifier to overfit on the few samples of the minority class. Table 1 summarizes the sampling techniques we studied.

2.3 Word Representation

As neural networks can only be applied to numeric input, the input token must be mapped to numerical values. One possibility would be a simple one-hot encoding, i.e., given a vocabulary of size n, the ith word is represented by the vector $[x_1, \ldots, x_n]$ with $x_i = 1$ and $x_j = 0$ for $j \neq i$. This encoding has two crucial disadvantages: The size of the representation is very big, and any linguistic information about the relations of different tokens is lost, which is why we are using a word embedding to map tokens into a vector space.

Tokens are represented using a 64 dimensional word embedding created using the word2vec approach by [Mi13]. We used the continuous skip-gram model for creating the word2vec model, i.e., the model has learned to predict tokens which are likely to appear in the same context as another token. In the final vector representation, words with similar meaning are mapped to vectors which are close to each other, which enables the neural network to detect errors in contexts it has not seen before.

All tokens which appeared at least five times in the training corpus are part of the word2vec model's dictionary. This way, the model is kept small by ignoring less frequently used tokens, and possible typos in the training corpus, which probably do not occur very often. Tokens which are not part of the dictionary are replaced by the special token "UNKNOWN". To minimize storage space, the same embedding is used for each confusion pair.

2.4 Neural Network Architecture

The artificial neural network gets the two tokens before and after a confusion word candidate as input. It outputs a number y_i for each token in the confusion set, which can be interpreted as the logits, i.e., the logarithm of the odd $\log\left(\frac{p}{1-p}\right)$, where p is the probability for the corresponding token to be correct in the given context.

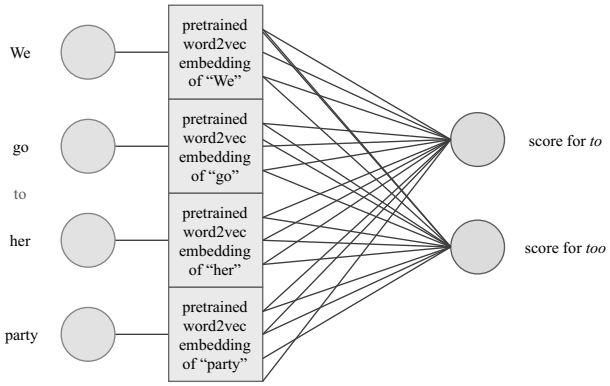


Fig. 1: An illustration of our "NN" architecture, which uses a separately trained word2vec embedding and no further hidden layers, for the confusion set {to, too}.

Our main architecture is a single layer network without any hidden layers and activation function, i.e., a linear model (called "NN", depicted in figure 1). For comparison, we also trained a network with one hidden layer with 8 neurons and ReLU activation function ("NNH") and variants which get only two tokens from the context as input ("NN2" and "NNH2", respectively). We did not train any deep models or models with large hidden layers because our goal was to create a classifier which does not require much storage memory.

All architectures are trained for 1.000 epochs using the Adam Optimizer by Kingma; Ba [KB14] to minimize the softmax cross entropy loss

$$L = -\log\left(\frac{e^{y_i}}{e^{y_i} + e^{y_j}}\right) \tag{1}$$

where y_i is the output for the correct label and y_j the output for the wrong label.

2.5 Output Interpretation

The output (y, y') of the neural network is used as follows: Given a threshold $\theta \in \mathbb{R}^+$, the token t of the confusion set is considered incorrect and t' is considered correct, if and only if $y < -\theta$ and $y' > \theta$ (i.e., the network thinks t' is much more likely than t and t' seems to fit).

The reasonableness of this approach can be explained like this: Assuming that in a given context the probabilities for t and t' are independent, i.e., it is possible that both tokens are correct, the output (y, y') can be transformed into probabilities using the sigmoid function. So we assume that

$$p_t = \frac{1}{1 + e^{-y}} \qquad p_{t'} = \frac{1}{1 + e^{-y'}} \tag{2}$$

are the probabilities that the first or second token are correct, respectively. Then the aforementioned approach is equivalent to saying that $p_t < 0.5 + \sigma$ and $p'_t > 0.5 - \sigma$, with

$$\sigma = \frac{1}{1 + e^{-\theta}} - 0.5 \in [0, 0.5) \tag{3}$$

i.e., t' is considered at least 2σ more probable to be correct than t and $p_t < 0.5$ and $p_{t'} > 0.5$.

The practical advantage of the first criterion is that it requires fewer calculations, and is therefore used in our implementation.

3 Rule Quality and Comparison

In this section we will have a look at the quality of the rules with regard to precision and recall, comparing our different architectures and the existing 3-gram-based rules.

3.1 Precision and Recall

In order to be useful for a grammar checking application, the neural network based rules must not cause any or at least very few false alarms. In the context of the error detection task, we define true positives (tp), true negatives (tn), false positives (fp) and false negatives (fn) as depicted in table 2.

Note that a true positive is an incorrect usage of a token which is marked as error, and not solely the case where the neural network would choose the right token (which is $tp + tn$).

	marked as error	not marked as error
correct usage	fp	tn
incorrect usage	tp	fn

Tab. 2: Definition of true positives, true negatives, false positives, false negatives. Note that a true positive is a wrong token correctly detected as wrong, and not the case where the neural network would insert the correct token.

For each confusion pair (t, t') we evaluated, we created a grammar checker rule and checked it against 5,000 sentences containing t and 5,000 sentences containing t' from the test set, and another 10,000 wrong sentences which were created by swapping t and t' in the correct sentences. We calculated precision P and recall R for different thresholds θ.

$$P = \frac{tp}{tp + fp} \qquad R = \frac{tp}{tp + fn} \tag{4}$$

A rule is considered good if $P > 0.99$ (i.e., the probability for false alarms is less than 1 %) and $R > 0.5$ (i.e., more than 50 % of incorrect usages are detected as error). For comparison: The average recall of the existing 3-gram rules for English in LanguageTool is 0.56 with $P > 0.99$.

3.2 Comparison of network architectures

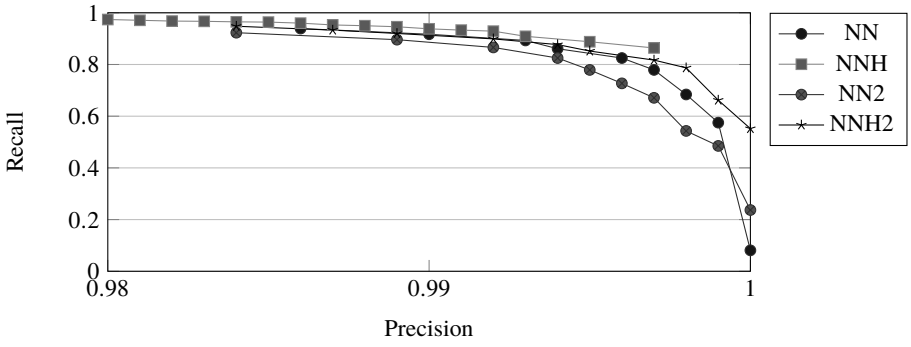


Fig. 2: Precision and recall for different network architectures for the confusion pair {to, too}

The neural network architectures show different recalls at the same level of precision on the test corpus. In general, looking at different confusion pairs, the architectures having 2 tokens as input have for a fixed precision lower recall than the corresponding architecture with 4 input tokens.

Moreover, the architectures with hidden layer perform better than those without hidden layer. Whether "NN" or "NNH2" performed better depended on the confusion set. The distance between the smaller "NN" architecture and the larger "NNH" within the interesting precision interval $[0.99, 0.995]$ has, in general, been rather small.

3.3 Comparison of Sampling Techniques

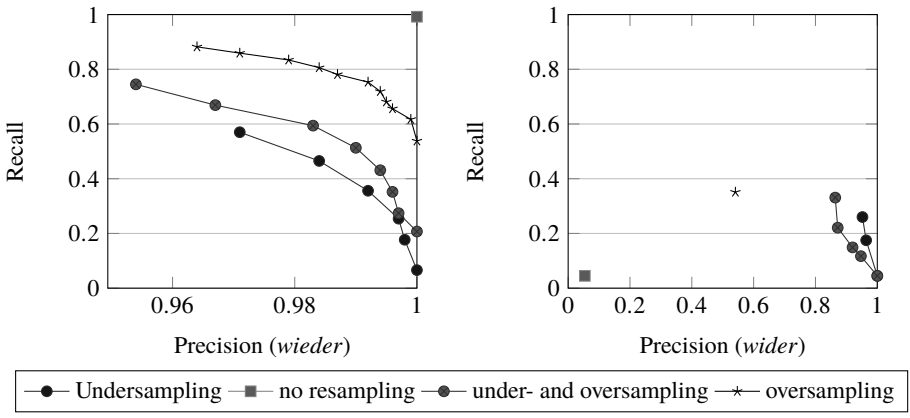


Fig. 3: Precision and recall for the confusion pair {wieder, wider}

We also had a look on how different sampling methods during the training process influenced the performance on the test set. Figure 3 shows precision and recall for the {wider, wieder} confusion pair using the "NN" architecture. For the diagram for *wieder*, only sentences where *wieder* is correct has been used, i.e., sentences with correct usage of *wieder* and sentences with incorrect usage of *wider*.

While there are decent results for detecting the right use of the more common word *wieder* when oversampling is used, the recall for the around 100 times less common *wider* is much worse, with a maximum precision of around 0.5, probably due to overfitting. If no resampling is used, the network is very good at dealing with contexts where *wieder* is correct, but has very low precision in contexts where *wider* must be used. Using a mixture of over- and undersampling produces relatively close results, where undersampling is worse for the recall of the more common *wieder* case and better for the less common *wider*.

For other imbalanced confusion pairs like {to, too} (factor 10), the differences have not been that big, such that undersampling has been used in the other experiments.

3.4 Comparison with 3-gram Rules

As our goal was to be at least as good as the existing 3-gram rules, we also compared the performance of our system with the existing rules. Note, however, that the comparison is not

confusion pair	$R_{3\text{-gram}}$	R_{NN}
and/end	0.81	**0.84**
five/give	**0.97**	0.94
it/its	**0.95**	0.92
our/out	**0.98**	0.93
then/the	0.45	**0.57**
to/too	0.82	**0.95**
some/same	**0.99**	0.98

confusion pair	$R_{3\text{-gram}}$	R_{NN}
da/dar	**0.82**	0.74
das/dass	0.43	**0.81**
den/denn	0.70	**0.90**
fielen/vielen	0.85	**0.94**
ihm/im	**0.96**	0.94
schon/schön	**0.73**	0.44
seid/seit	**0.98**	0.93

Tab. 3: Comparison of recall at $P = 0.99$ for some English and German confusion pairs.

completely accurate, since the 3-gram rules use a different tokenization algorithm, which is compatible with Google's n-gram database. For instance, the 3-gram rules can detect the error in **give-year-old*, because this expression consists of 5 tokens according to the Google style tokenizer, whereas our rules fail to detect the error, since the expression is one token for the LanguageTool tokenizer. In order not to end up with a lot of "false" false negatives for our rules, we changed the existing testing algorithm in LanguageTool to exclude those cases.

The results depicted in table 3 show that our rules have, on average, a performance comparable to those using the memory-based 3-gram rules. In around half of the cases, our simple one-layer architecture outperforms the 3-gram rules.

3.5 Memory Usage and Runtime Performance

The files for the word2vec embedding for English have a size of around 65 MB (uncompressed, stored as plain text data). The files containing the neural network weights for the "NN" architecture have a size of 13 KB for each confusion pair. Thus, around 800,000 neural network based rules would need the same amount of storage memory as the 3-gram corpus, which is stored as 11 GB Lucene database index.

The start-up wall-clock time of the LanguageTool standalone GUI without 3-gram and neural network rules, from the start till an English example sentence has been checked against the intergrated pattern rules, is about 4.8 seconds on our test system with SSD. If only the 3-gram rules (and the pattern rules) are enabled, the start-up time is 1.2 seconds longer, with only neural network rules (and pattern rules) enabled, the time is 1.5 seconds longer.

The memory usage of the GUI 10 seconds after start-up and a garbage collection call is around 80 MB without 3-gram and neural network rules, 130 MB with 3-gram rules enabled and 100 MB with neural network rules loaded.

Checking a German text with around 3,000 words using the command line version of LanguageTool takes around 2.9 seconds with both rule types disabled and only pattern rules enabled, 4.5 seconds with 76 3-gram rules enabled and 3.0 seconds with 29 neural network rules of the "NN" architecture enabled. Hence we can see that our approach has a much lower impact on the time of the grammar checking process.

To sum up, the calculation done by the neural network code have a lower impact on the performance than the 3-gram lookup, and storing as well as loading the 3-gram index requires more memory than the word2vec model and the neural network data. Only the start-up time of LanguageTool is slightly negatively affected.

4 Alternative Approaches

During development of the architectures discussed in the previous section, we also had a look at other architectures and classifiers. In this section, we summarize which other approaches for the confusion pair disambiguation task have also been tested, and why we think that those approaches are inferior.

4.1 Classical Machine Learning

We also did experiments with classical machine learning classifiers which also got word2vec encoded context words as input.

Random forest are fast to train, had a precision of over 85% for most confusion pairs, but were unable to reach our target of 99% without further optimization. Another drawback is their model size of around 1.5 MB for a random forest with 20 decision trees, which is 100 more than needed by the "NN" architecture. Furthermore, random forests were considerably slower at evaluation time.

On the other hand, Support Vector Machines (SVMs) required more time at training time, were fast at test time, and reached results comparable with those for the "NN" architecture. But like random forests, the model sizes were larger than 1.5 MB, which was the main reason why we did not further evaluate SVMs.

4.2 Recurrent and Convolutional Neural Networks

As recurrent and convolutional neural networks are able to handle inputs of different lengths, they can easily be used to analyze sentences of different lengths. Although it seems promising to use these architectures, there are several drawbacks. First, the model size is much bigger because the model must be able to deal with larger inputs. It also takes more

time to check a sentence, because more computations have to be done, which can slow down the performance of the grammar checker considerably, especially if no GPU can be used. Furthermore, especially the training of recurrent neural networks takes a considerable amount of time, which would make rule creation a very time-consuming task.

5 Related Work

Miłkowski [Mi12] has studied automatic and semi-automatic creation of symbolic rules using transformation-based learning. The created rules have very good recall, but often suffer from a low precision, i.e., cause many false alarms, unless there is human intervention.

Support vector machines, convolution neural networks with fixed context size and recurrent neural networks for detecting grammar errors at the word level using unlabeled data have been compared by Liu; Liu [LL17]. A bidirectional LSTM-based classifier performed best, but still has an F-measure below 20% which makes it unsuitable for application in a grammar checker because it would cause too many false alarms. Because of that, we focused on the simpler task of confusion set disambiguation.

Banko; Brill [BB01] have compared different classifiers for the confusion set task with regard to their performance if the training corpus is increased from 1 million words to 1 billion words. They have shown that a memory based algorithm is outperformed by a more complex perceptron algorithm when the training corpus has more than 1 million words.

6 Conclusion and Future Work

In this paper we have presented a new kind of rule for the free style and grammar checker LanguageTool which uses neural networks, and tested them successfully on a confusion set disambiguation task. The rule quality is similar to the memory based rules which are already part of LanguageTool, but our rules require less memory and are faster. Hence, our rule can be used instead or in addition to the existing 3-gram rules. It has to be noted, though, that creating new neural network based rules requires several minutes of computation time for the training process, which is not needed for a new 3-gram rule.

Possible next steps include using information from the part-of-speech tagger to handle words which are not part of the training vocabulary more appropriately than projecting all out-of-vocabulary tokens to a single “UNKNOWN” token. Furthermore, the current neural network architecture can easily be extended to support bigger confusion sets, such that rules for {to, too}, {to, two} and {two, too} can be merged in one {to, too, two} rule. Adding support for confusion sets containing larger expressions instead of single tokens (e.g. {das, dass,} or {in dem, indem}) is also planned. In addition, training on even larger corpora might further improve the performance. It is also possible to reduce the storage space for the neural network rules by using a binary format for storing weight matrices.

Moreover, adding support of new confusion pairs could be simplified by creating a neural network which is not specialized on a confusion pair, but can calculate probabilities for any target tokens.

Acknowledgements

Computational support and infrastructure was provided by the "Center for Information and Media Technology" (ZIM) at the University of Düsseldorf (Germany). I would like to thank my supervisor Sebastian Krings for his useful advises, and I also thank the LanguageTool community for the feedback during the integration of the new rules into LanguageTool.

References

[BB01] Banko, M.; Brill, E.: Scaling to very very large corpora for natural language disambiguation. In: Proceedings of the 39th annual meeting on association for computational linguistics. Association for Computational Linguistics, pp. 26–33, 2001.

[Ch09] Chawla, N. V.: Data mining for imbalanced datasets: An overview. In: Data mining and knowledge discovery handbook. Springer, pp. 875–886, 2009.

[GEQ12] Goldhahn, D.; Eckart, T.; Quasthoff, U.: Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. In: LREC. Pp. 759–765, 2012.

[Ho] Ho, T.: Tatoeba Downloads, URL: `https://tatoeba.org/eng/downloads`, visited on: 11/01/2017.

[KB14] Kingma, D.; Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980/, 2014.

[Li12] Lin, Y.; Michel, J.-B.; Aiden, E. L.; Orwant, J.; Brockman, W.; Petrov, S.: Syntactic annotations for the google books ngram corpus. In: Proceedings of the ACL 2012 system demonstrations. Association for Computational Linguistics, pp. 169–174, 2012.

[LL17] Liu, Z.-R.; Liu, Y.: Exploiting Unlabeled Data for Neural Grammatical Error Detection. Journal of Computer Science and Technology 32/4, pp. 758–767, 2017.

[Mi12] Miłkowski, M.: Automating rule generation for grammar checkers. arXiv preprint arXiv:1211.6887/, 2012.

[Mi13] Mikolov, T.; Chen, K.; Corrado, G.; Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781/, 2013.

[Na03] Naber, D.: A rule-based style and grammar checker./, 2003.

Autorenverzeichnis

GI-Edition Lecture Notes in Informatics

P-1 Gregor Engels, Andreas Oberweis, Albert Zündorf (Hrsg.): Modellierung 2001.

P-2 Mikhail Godlevsky, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications, ISTA'2001.

P-3 Ana M. Moreno, Reind P. van de Riet (Hrsg.): Applications of Natural Lan-guage to Information Systems, NLDB'2001.

P-4 H. Wörn, J. Mühling, C. Vahl, H.-P. Meinzer (Hrsg.): Rechner- und sensorgestützte Chirurgie; Workshop des SFB 414.

P-5 Andy Schürr (Hg.): OMER – Object-Oriented Modeling of Embedded Real-Time Systems.

P-6 Hans-Jürgen Appelrath, Rolf Beyer, Uwe Marquardt, Heinrich C. Mayr, Claudia Steinberger (Hrsg.): Unternehmen Hochschule, UH'2001.

P-7 Andy Evans, Robert France, Ana Moreira, Bernhard Rumpe (Hrsg.): Practical UML-Based Rigorous Development Methods – Countering or Integrating the extremists, pUML'2001.

P-8 Reinhard Keil-Slawik, Johannes Magenheim (Hrsg.): Informatikunterricht und Medienbildung, INFOS'2001.

P-9 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Innovative Anwendungen in Kommunikationsnetzen, 15. DFN Arbeitstagung.

P-10 Mirjam Minor, Steffen Staab (Hrsg.): 1st German Workshop on Experience Management: Sharing Experiences about the Sharing Experience.

P-11 Michael Weber, Frank Kargl (Hrsg.): Mobile Ad-Hoc Netzwerke, WMAN 2002.

P-12 Martin Glinz, Günther Müller-Luschnat (Hrsg.): Modellierung 2002.

P-13 Jan von Knop, Peter Schirmbacher and Viljan Mahni_ (Hrsg.): The Changing Universities – The Role of Technology.

P-14 Robert Tolksdorf, Rainer Eckstein (Hrsg.): XML-Technologien für das Semantic Web – XSW 2002.

P-15 Hans-Bernd Bludau, Andreas Koop (Hrsg.): Mobile Computing in Medicine.

P-16 J. Felix Hampe, Gerhard Schwabe (Hrsg.): Mobile and Collaborative Business 2002.

P-17 Jan von Knop, Wilhelm Haverkamp (Hrsg.): Zukunft der Netze –Die Verletzbarkeit meistern, 16. DFN Arbeitstagung.

P-18 Elmar J. Sinz, Markus Plaha (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2002.

P-19 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3. Okt. 2002 in Dortmund.

P-20 Sigrid Schubert, Bernd Reusch, Norbert Jesse (Hrsg.): Informatik bewegt – Informatik 2002 – 32. Jahrestagung der Gesellschaft für Informatik e.V. (GI) 30.Sept.-3. Okt. 2002 in Dortmund (Ergänzungsband).

P-21 Jörg Desel, Mathias Weske (Hrsg.): Promise 2002: Prozessorientierte Methoden und Werkzeuge für die Entwicklung von Informationssystemen.

P-22 Sigrid Schubert, Johannes Magenheim, Peter Hubwieser, Torsten Brinda (Hrsg.): Forschungsbeiträge zur "Didaktik der Informatik" – Theorie, Praxis, Evaluation.

P-23 Thorsten Spitta, Jens Borchers, Harry M. Sneed (Hrsg.): Software Management 2002 – Fortschritt durch Beständigkeit

P-24 Rainer Eckstein, Robert Tolksdorf (Hrsg.): XMIDX 2003 – XML-Technologien für Middleware – Middleware für XML-Anwendungen

P-25 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Commerce – Anwendungen und Perspektiven – 3. Workshop Mobile Commerce, Universität Augsburg, 04.02.2003

P-26 Gerhard Weikum, Harald Schöning, Erhard Rahm (Hrsg.): BTW 2003: Datenbanksysteme für Business, Technologie und Web

P-27 Michael Kroll, Hans-Gerd Lipinski, Kay Melzer (Hrsg.): Mobiles Computing in der Medizin

P-28 Ulrich Reimer, Andreas Abecker, Steffen Staab, Gerd Stumme (Hrsg.): WM 2003: Professionelles Wissensmanagement – Er-fahrungen und Visionen

P-29 Antje Düsterhöft, Bernhard Thalheim (Eds.): NLDB'2003: Natural Language Processing and Information Systems

P-30 Mikhail Godlevsky, Stephen Liddle, Heinrich C. Mayr (Eds.): Information Systems Technology and its Applications

P-31 Arslan Brömme, Christoph Busch (Eds.): BIOSIG 2003: Biometrics and Electronic Signatures

P-32 Peter Hubwieser (Hrsg.): Informatische Fachkonzepte im Unterricht – INFOS 2003

P-33 Andreas Geyer-Schulz, Alfred Taudes (Hrsg.): Informationswirtschaft: Ein Sektor mit Zukunft

P-34 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 1)

P-35 Klaus Dittrich, Wolfgang König, Andreas Oberweis, Kai Rannenberg, Wolfgang Wahlster (Hrsg.): Informatik 2003 – Innovative Informatikanwendungen (Band 2)

P-36 Rüdiger Grimm, Hubert B. Keller, Kai Rannenberg (Hrsg.): Informatik 2003 – Mit Sicherheit Informatik

P-37 Arndt Bode, Jörg Desel, Sabine Rathmayer, Martin Wessner (Hrsg.): DeLFI 2003: e-Learning Fachtagung Informatik

P-38 E.J. Sinz, M. Plaha, P. Neckel (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2003

P-39 Jens Nedon, Sandra Frings, Oliver Göbel (Hrsg.): IT-Incident Management & IT-Forensics – IMF 2003

P-40 Michael Rebstock (Hrsg.): Modellierung betrieblicher Informationssysteme – MobIS 2004

P-41 Uwe Brinkschulte, Jürgen Becker, Dietmar Fey, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle, Thomas Runkler (Edts.): ARCS 2004 – Organic and Pervasive Computing

P-42 Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Economy – Transaktionen und Prozesse, Anwendungen und Dienste

P-43 Birgitta König-Ries, Michael Klein, Philipp Obreiter (Hrsg.): Persistance, Scalability, Transactions – Database Mechanisms for Mobile Applications

P-44 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): Security, E-Learning. E-Services

P-45 Bernhard Rumpe, Wofgang Hesse (Hrsg.): Modellierung 2004

P-46 Ulrich Flegel, Michael Meier (Hrsg.): Detection of Intrusions of Malware & Vulnerability Assessment

P-47 Alexander Prosser, Robert Krimmer (Hrsg.): Electronic Voting in Europe – Technology, Law, Politics and Society

P-48 Anatoly Doroshenko, Terry Halpin, Stephen W. Liddle, Heinrich C. Mayr (Hrsg.): Information Systems Technology and its Applications

P-49 G. Schiefer, P. Wagner, M. Morgenstern, U. Rickert (Hrsg.): Integration und Datensicherheit – Anforderungen, Konflikte und Perspektiven

P-50 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 1) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm

P-51 Peter Dadam, Manfred Reichert (Hrsg.): INFORMATIK 2004 – Informatik verbindet (Band 2) Beiträge der 34. Jahrestagung der Gesellschaft für Informatik e.V. (GI), 20.-24. September 2004 in Ulm

P-52 Gregor Engels, Silke Seehusen (Hrsg.): DELFI 2004 – Tagungsband der 2. e-Learning Fachtagung Informatik

P-53 Robert Giegerich, Jens Stoye (Hrsg.): German Conference on Bioinformatics – GCB 2004

P-54 Jens Borchers, Ralf Kneuper (Hrsg.): Softwaremanagement 2004 – Outsourcing und Integration

P-55 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): E-Science und Grid Adhoc-Netze Medienintegration

P-56 Fernand Feltz, Andreas Oberweis, Benoit Otjacques (Hrsg.): EMISA 2004 – Informationssysteme im E-Business und E-Government

P-57 Klaus Turowski (Hrsg.): Architekturen, Komponenten, Anwendungen

P-58 Sami Beydeda, Volker Gruhn, Johannes Mayer, Ralf Reussner, Franz Schweiggert (Hrsg.): Testing of Component-Based Systems and Software Quality

P-59 J. Felix Hampe, Franz Lehner, Key Pousttchi, Kai Ranneberg, Klaus Turowski (Hrsg.): Mobile Business – Processes, Platforms, Payments

P-60 Steffen Friedrich (Hrsg.): Unterrichtskonzepte für inforrmatische Bildung

P-61 Paul Müller, Reinhard Gotzhein, Jens B. Schmitt (Hrsg.): Kommunikation in verteilten Systemen

P-62 Federrath, Hannes (Hrsg.): „Sicherheit 2005“ – Sicherheit – Schutz und Zuverlässigkeit

P-63 Roland Kaschek, Heinrich C. Mayr, Stephen Liddle (Hrsg.): Information Systems – Technology and ist Applications

P-64 Peter Liggesmeyer, Klaus Pohl, Michael Goedicke (Hrsg.): Software Engineering 2005

P-65 Gottfried Vossen, Frank Leymann, Peter Lockemann, Wolffried Stucky (Hrsg.): Datenbanksysteme in Business, Technologie und Web

P-66 Jörg M. Haake, Ulrike Lucke, Djamshid Tavangarian (Hrsg.): DeLFI 2005: 3. deutsche e-Learning Fachtagung Informatik

P-67 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 1)

P-68 Armin B. Cremers, Rainer Manthey, Peter Martini, Volker Steinhage (Hrsg.): INFORMATIK 2005 – Informatik LIVE (Band 2)

P-69 Robert Hirschfeld, Ryszard Kowalcyk, Andreas Polze, Matthias Weske (Hrsg.): NODe 2005, GSEM 2005

P-70 Klaus Turowski, Johannes-Maria Zaha (Hrsg.): Component-oriented Enterprise Application (COAE 2005)

P-71 Andrew Torda, Stefan Kurz, Matthias Rarey (Hrsg.): German Conference on Bioinformatics 2005

P-72 Klaus P. Jantke, Klaus-Peter Fähnrich, Wolfgang S. Wittig (Hrsg.): Marktplatz Internet: Von e-Learning bis e-Payment

P-73 Jan von Knop, Wilhelm Haverkamp, Eike Jessen (Hrsg.): “Heute schon das Morgen sehen“

P-74 Christopher Wolf, Stefan Lucks, Po-Wah Yau (Hrsg.): WEWoRC 2005 – Western European Workshop on Research in Cryptology

P-75 Jörg Desel, Ulrich Frank (Hrsg.): Enterprise Modelling and Information Systems Architecture

P-76 Thomas Kirste, Birgitta König-Riess, Key Pousttchi, Klaus Turowski (Hrsg.): Mobile Informationssysteme – Potentiale, Hindernisse, Einsatz

P-77 Jana Dittmann (Hrsg.): SICHERHEIT 2006

P-78 K.-O. Wenkel, P. Wagner, M. Morgenstern, K. Luzi, P. Eisermann (Hrsg.): Land- und Ernährungswirtschaft im Wandel

P-79 Bettina Biel, Matthias Book, Volker Gruhn (Hrsg.): Softwareengineering 2006

P-80 Mareike Schoop, Christian Huemer, Michael Rebstock, Martin Bichler (Hrsg.): Service-Oriented Electronic Commerce

P-81 Wolfgang Karl, Jürgen Becker, Karl-Erwin Großpietsch, Christian Hochberger, Erik Maehle (Hrsg.): ARCS´06

P-82 Heinrich C. Mayr, Ruth Breu (Hrsg.): Modellierung 2006

P-83 Daniel Huson, Oliver Kohlbacher, Andrei Lupas, Kay Nieselt and Andreas Zell (eds.): German Conference on Bioinformatics

P-84 Dimitris Karagiannis, Heinrich C. Mayr, (Hrsg.): Information Systems Technology and its Applications

P-85 Witold Abramowicz, Heinrich C. Mayr, (Hrsg.): Business Information Systems

P-86 Robert Krimmer (Ed.): Electronic Voting 2006

P-87 Max Mühlhäuser, Guido Rößling, Ralf Steinmetz (Hrsg.): DELFI 2006: 4. e-Learning Fachtagung Informatik

P-88 Robert Hirschfeld, Andreas Polze, Ryszard Kowalczyk (Hrsg.): NODe 2006, GSEM 2006

P-90 Joachim Schelp, Robert Winter, Ulrich Frank, Bodo Rieger, Klaus Turowski (Hrsg.): Integration, Informationslogistik und Architektur

P-91 Henrik Stormer, Andreas Meier, Michael Schumacher (Eds.): European Conference on eHealth 2006

P-92 Fernand Feltz, Benoît Otjacques, Andreas Oberweis, Nicolas Poussing (Eds.): AIM 2006

P-93 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 1

P-94 Christian Hochberger, Rüdiger Liskowsky (Eds.): INFORMATIK 2006 – Informatik für Menschen, Band 2

P-95 Matthias Weske, Markus Nüttgens (Eds.): EMISA 2005: Methoden, Konzepte und Technologien für die Entwicklung von dienstbasierten Informationssystemen

P-96 Saartje Brockmans, Jürgen Jung, York Sure (Eds.): Meta-Modelling and Ontologies

P-97 Oliver Göbel, Dirk Schadt, Sandra Frings, Hardo Hase, Detlef Günther, Jens Nedon (Eds.): IT-Incident Mangament & IT-Forensics – IMF 2006

P-98 Hans Brandt-Pook, Werner Simonsmeier und Thorsten Spitta (Hrsg.): Beratung in der Softwareentwicklung – Modelle, Methoden, Best Practices

P-99 Andreas Schwill, Carsten Schulte, Marco Thomas (Hrsg.): Didaktik der Informatik

P-100 Peter Forbrig, Günter Siegel, Markus Schneider (Hrsg.): HDI 2006: Hochschuldidaktik der Informatik

P-101 Stefan Böttinger, Ludwig Theuvsen, Susanne Rank, Marlies Morgenstern (Hrsg.): Agrarinformatik im Spannungsfeld zwischen Regionalisierung und globalen Wertschöpfungsketten

P-102 Otto Spaniol (Eds.): Mobile Services and Personalized Environments

P-103 Alfons Kemper, Harald Schöning, Thomas Rose, Matthias Jarke, Thomas Seidl, Christoph Quix, Christoph Brochhaus (Hrsg.): Datenbanksysteme in Business, Technologie und Web (BTW 2007)

P-104 Birgitta König-Ries, Franz Lehner, Rainer Malaka, Can Türker (Hrsg.) MMS 2007: Mobilität und mobile Informationssysteme

P-105 Wolf-Gideon Bleek, Jörg Raasch, Heinz Züllighoven (Hrsg.) Software Engineering 2007

P-106 Wolf-Gideon Bleek, Henning Schwentner, Heinz Züllighoven (Hrsg.) Software Engineering 2007 – Beiträge zu den Workshops

P-107 Heinrich C. Mayr, Dimitris Karagiannis (eds.) Information Systems Technology and its Applications

P-108 Arslan Brömme, Christoph Busch, Detlef Hühnlein (eds.) BIOSIG 2007: Biometrics and Electronic Signatures

P-109 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 1

P-110 Rainer Koschke, Otthein Herzog, Karl-Heinz Rödiger, Marc Ronthaler (Hrsg.) INFORMATIK 2007 Informatik trifft Logistik Band 2

P-111 Christian Eibl, Johannes Magenheim, Sigrid Schubert, Martin Wessner (Hrsg.) DeLFI 2007: 5. e-Learning Fachtagung Informatik

P-112 Sigrid Schubert (Hrsg.) Didaktik der Informatik in Theorie und Praxis

P-113 Sören Auer, Christian Bizer, Claudia Müller, Anna V. Zhdanova (Eds.) The Social Semantic Web 2007 Proceedings of the 1st Conference on Social Semantic Web (CSSW)

P-114 Sandra Frings, Oliver Göbel, Detlef Günther, Hardo G. Hase, Jens Nedon, Dirk Schadt, Arslan Brömme (Eds.) IMF2007 IT-incident management & IT-forensics Proceedings of the 3rd International Conference on IT-Incident Management & IT-Forensics

P-115 Claudia Falter, Alexander Schliep, Joachim Selbig, Martin Vingron and Dirk Walther (Eds.) German conference on bioinformatics GCB 2007

P-116 Witold Abramowicz, Leszek Maciszek (Eds.) Business Process and Services Computing 1st International Working Conference on Business Process and Services Computing BPSC 2007

P-117 Ryszard Kowalczyk (Ed.) Grid service engineering and manegement The 4th International Conference on Grid Service Engineering and Management GSEM 2007

P-118 Andreas Hein, Wilfried Thoben, Hans-Jürgen Appelrath, Peter Jensch (Eds.) European Conference on ehealth 2007

P-119 Manfred Reichert, Stefan Strecker, Klaus Turowski (Eds.) Enterprise Modelling and Information Systems Architectures Concepts and Applications

P-120 Adam Pawlak, Kurt Sandkuhl, Wojciech Cholewa, Leandro Soares Indrusiak (Eds.) Coordination of Collaborative Engineering - State of the Art and Future Challenges

P-121 Korbinian Herrmann, Bernd Bruegge (Hrsg.) Software Engineering 2008 Fachtagung des GI-Fachbereichs Softwaretechnik

P-122 Walid Maalej, Bernd Bruegge (Hrsg.) Software Engineering 2008 - Workshopband Fachtagung des GI-Fachbereichs Softwaretechnik

P-123 Michael H. Breitner, Martin Breunig, Elgar Fleisch, Ley Pousttchi, Klaus Turowski (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Technologien, Prozesse, Marktfähigkeit
Proceedings zur 3. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2008)

P-124 Wolfgang E. Nagel, Rolf Hoffmann, Andreas Koch (Eds.)
9th Workshop on Parallel Systems and Algorithms (PASA)
Workshop of the GI/ITG Speciel Interest Groups PARS and PARVA

P-125 Rolf A.E. Müller, Hans-H. Sundermeier, Ludwig Theuvsen, Stephanie Schütze, Marlies Morgenstern (Hrsg.)
Unternehmens-IT:
Führungsinstrument oder Verwaltungsbürde
Referate der 28. GIL Jahrestagung

P-126 Rainer Gimnich, Uwe Kaiser, Jochen Quante, Andreas Winter (Hrsg.)
10th Workshop Software Reengineering (WSR 2008)

P-127 Thomas Kühne, Wolfgang Reisig, Friedrich Steimann (Hrsg.)
Modellierung 2008

P-128 Ammar Alkassar, Jörg Siekmann (Hrsg.)
Sicherheit 2008
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 4. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)
2.-4. April 2008
Saarbrücken, Germany

P-129 Wolfgang Hesse, Andreas Oberweis (Eds.)
Sigsand-Europe 2008
Proceedings of the Third AIS SIGSAND European Symposium on Analysis, Design, Use and Societal Impact of Information Systems

P-130 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
1. DFN-Forum Kommunikations-technologien Beiträge der Fachtagung

P-131 Robert Krimmer, Rüdiger Grimm (Eds.)
3rd International Conference on Electronic Voting 2008
Co-organized by Council of Europe, Gesellschaft für Informatik and E-Voting. CC

P-132 Silke Seehusen, Ulrike Lucke, Stefan Fischer (Hrsg.)
DeLFI 2008:
Die 6. e-Learning Fachtagung Informatik

P-133 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik
Band 1

P-134 Heinz-Gerd Hegering, Axel Lehmann, Hans Jürgen Ohlbach, Christian Scheideler (Hrsg.)
INFORMATIK 2008
Beherrschbare Systeme – dank Informatik
Band 2

P-135 Torsten Brinda, Michael Fothe, Peter Hubwieser, Kirsten Schlüter (Hrsg.)
Didaktik der Informatik –
Aktuelle Forschungsergebnisse

P-136 Andreas Beyer, Michael Schroeder (Eds.)
German Conference on Bioinformatics GCB 2008

P-137 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)
BIOSIG 2008: Biometrics and Electronic Signatures

P-138 Barbara Dinter, Robert Winter, Peter Chamoni, Norbert Gronau, Klaus Turowski (Hrsg.)
Synergien durch Integration und Informationslogistik
Proceedings zur DW2008

P-139 Georg Herzwurm, Martin Mikusz (Hrsg.)
Industrialisierung des Software-Managements
Fachtagung des GI-Fachausschusses Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschaftsinformatik

P-140 Oliver Göbel, Sandra Frings, Detlef Günther, Jens Nedon, Dirk Schadt (Eds.)
IMF 2008 - IT Incident Management & IT Forensics

P-141 Peter Loos, Markus Nüttgens, Klaus Turowski, Dirk Werth (Hrsg.)
Modellierung betrieblicher Informations-systeme (MobIS 2008)
Modellierung zwischen SOA und Compliance Management

P-142 R. Bill, P. Korduan, L. Theuvsen, M. Morgenstern (Hrsg.)
Anforderungen an die Agrarinformatik durch Globalisierung und Klimaveränderung

P-143 Peter Liggesmeyer, Gregor Engels, Jürgen Münch, Jörg Dörr, Norman Riegel (Hrsg.)
Software Engineering 2009
Fachtagung des GI-Fachbereichs Softwaretechnik

P-144 Johann-Christoph Freytag, Thomas Ruf, Wolfgang Lehner, Gottfried Vossen (Hrsg.)
Datenbanksysteme in Business, Technologie und Web (BTW)

P-145 Knut Hinkelmann, Holger Wache (Eds.)
WM2009: 5th Conference on Professional Knowledge Management

P-146 Markus Bick, Martin Breunig, Hagen Höpfner (Hrsg.)
Mobile und Ubiquitäre Informationssysteme – Entwicklung, Implementierung und Anwendung
4. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2009)

P-147 Witold Abramowicz, Leszek Maciaszek, Ryszard Kowalczyk, Andreas Speck (Eds.)
Business Process, Services Computing and Intelligent Service Management
BPSC 2009 · ISM 2009 · YRW-MBP 2009

P-148 Christian Erfurth, Gerald Eichler, Volkmar Schau (Eds.)
9th International Conference on Innovative Internet Community Systems
I^{2}CS 2009

P-149 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
2. DFN-Forum Kommunikationstechnologien
Beiträge der Fachtagung

P-150 Jürgen Münch, Peter Liggesmeyer (Hrsg.)
Software Engineering
2009 - Workshopband

P-151 Armin Heinzl, Peter Dadam, Stefan Kirn, Peter Lockemann (Eds.)
PRIMIUM
Process Innovation for Enterprise Software

P-152 Jan Mendling, Stefanie Rinderle-Ma, Werner Esswein (Eds.)
Enterprise Modelling and Information Systems Architectures
Proceedings of the 3rd Int'l Workshop EMISA 2009

P-153 Andreas Schwill, Nicolas Apostolopoulos (Hrsg.)
Lernen im Digitalen Zeitalter
DeLFI 2009 – Die 7. E-Learning Fachtagung Informatik

P-154 Stefan Fischer, Erik Maehle Rüdiger Reischuk (Hrsg.)
INFORMATIK 2009
Im Focus das Leben

P-155 Arslan Brömme, Christoph Busch, Detlef Hühnlein (Eds.)
BIOSIG 2009:
Biometrics and Electronic Signatures
Proceedings of the Special Interest Group on Biometrics and Electronic Signatures

P-156 Bernhard Koerber (Hrsg.)
Zukunft braucht Herkunft
25 Jahre »INFOS – Informatik und Schule«

P-157 Ivo Grosse, Steffen Neumann, Stefan Posch, Falk Schreiber, Peter Stadler (Eds.)
German Conference on Bioinformatics 2009

P-158 W. Claupein, L. Theuvsen, A. Kämpf, M. Morgenstern (Hrsg.)
Precision Agriculture Reloaded – Informationsgestützte Landwirtschaft

P-159 Gregor Engels, Markus Luckey, Wilhelm Schäfer (Hrsg.)
Software Engineering 2010

P-160 Gregor Engels, Markus Luckey, Alexander Pretschner, Ralf Reussner (Hrsg.)
Software Engineering 2010 – Workshopband
(inkl. Doktorandensymposium)

P-161 Gregor Engels, Dimitris Karagiannis Heinrich C. Mayr (Hrsg.)
Modellierung 2010

P-162 Maria A. Wimmer, Uwe Brinkhoff, Siegfried Kaiser, Dagmar Lück-Schneider, Erich Schweighofer, Andreas Wiebe (Hrsg.)
Vernetzte IT für einen effektiven Staat
Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2010

P-163 Markus Bick, Stefan Eulgem, Elgar Fleisch, J. Felix Hampe, Birgitta König-Ries, Franz Lehner, Key Pousttchi, Kai Rannenberg (Hrsg.)
Mobile und Ubiquitäre Informationssysteme
Technologien, Anwendungen und Dienste zur Unterstützung von mobiler Kollaboration

P-164 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2010: Biometrics and Electronic Signatures Proceedings of the Special Interest Group on Biometrics and Electronic Signatures

P-165 Gerald Eichler, Peter Kropf, Ulrike Lechner, Phayung Meesad, Herwig Unger (Eds.)
10th International Conference on Innovative Internet Community Systems (I^2CS) – Jubilee Edition 2010 –

P-166 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
3. DFN-Forum Kommunikationstechnologien
Beiträge der Fachtagung

P-167 Robert Krimmer, Rüdiger Grimm (Eds.)
4th International Conference on Electronic Voting 2010
co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC

P-168 Ira Diethelm, Christina Dörge, Claudia Hildebrandt, Carsten Schulte (Hrsg.)
Didaktik der Informatik
Möglichkeiten empirischer Forschungsmethoden und Perspektiven der Fachdidaktik

P-169 Michael Kerres, Nadine Ojstersek Ulrik Schroeder, Ulrich Hoppe (Hrsg.)
DeLFI 2010 - 8. Tagung der Fachgruppe E-Learning der Gesellschaft für Informatik e.V.

P-170 Felix C. Freiling (Hrsg.)
Sicherheit 2010
Sicherheit, Schutz und Zuverlässigkeit

P-171 Werner Esswein, Klaus Turowski, Martin Juhrisch (Hrsg.)
Modellierung betrieblicher Informationssysteme (MobIS 2010)
Modellgestütztes Management

P-172 Stefan Klink, Agnes Koschmider Marco Mevius, Andreas Oberweis (Hrsg.)
EMISA 2010
Einflussfaktoren auf die Entwicklung flexibler, integrierter Informationssysteme
Beiträge des Workshops der GI-Fachgruppe EMISA (Entwicklungsmethoden für Informationssysteme und deren Anwendung)

P-173 Dietmar Schomburg, Andreas Grote (Eds.)
German Conference on Bioinformatics 2010

P-174 Arslan Brömme, Torsten Eymann, Detlef Hühnlein, Heiko Roßnagel, Paul Schmücker (Hrsg.)
perspeGKtive 2010
Workshop „Innovative und sichere Informationstechnologie für das Gesundheitswesen von morgen“

P-175 Klaus-Peter Fähnrich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010
Service Science – Neue Perspektiven für die Informatik
Band 1

P-176 Klaus-Peter Fähnrich, Bogdan Franczyk (Hrsg.)
INFORMATIK 2010
Service Science – Neue Perspektiven für die Informatik
Band 2

P-177 Witold Abramowicz, Rainer Alt, Klaus-Peter Fähnrich, Bogdan Franczyk, Leszek A. Maciaszek (Eds.)
INFORMATIK 2010
Business Process and Service Science – Proceedings of ISSS and BPSC

P-178 Wolfram Pietsch, Benedikt Krams (Hrsg.)
Vom Projekt zum Produkt
Fachtagung des GI-Fachausschusses Management der Anwendungsentwicklung und -wartung im Fachbereich Wirtschafts-informatik (WI-MAW), Aachen, 2010

P-179 Stefan Gruner, Bernhard Rumpe (Eds.)
FM+AM`2010
Second International Workshop on Formal Methods and Agile Methods

P-180 Theo Härder, Wolfgang Lehner, Bernhard Mitschang, Harald Schöning, Holger Schwarz (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW)
14. Fachtagung des GI-Fachbereichs „Datenbanken und Informationssysteme“ (DBIS)

P-181 Michael Clasen, Otto Schätzel, Brigitte Theuvsen (Hrsg.)
Qualität und Effizienz durch informationsgestützte Landwirtschaft, Fokus: Moderne Weinwirtschaft

P-182 Ronald Maier (Hrsg.)
6th Conference on Professional Knowledge Management
From Knowledge to Action

P-183 Ralf Reussner, Matthias Grund, Andreas Oberweis, Walter Tichy (Hrsg.)
Software Engineering 2011
Fachtagung des GI-Fachbereichs Softwaretechnik

P-184 Ralf Reussner, Alexander Pretschner, Stefan Jähnichen (Hrsg.)
Software Engineering 2011
Workshopband
(inkl. Doktorandensymposium)

P-185 Hagen Höpfner, Günther Specht, Thomas Ritz, Christian Bunse (Hrsg.)
MMS 2011: Mobile und ubiquitäre Informationssysteme Proceedings zur 6. Konferenz Mobile und Ubiquitäre Informationssysteme (MMS 2011)

P-186 Gerald Eichler, Axel Küpper, Volkmar Schau, Hacène Fouchal, Herwig Unger (Eds.)
11th International Conference on Innovative Internet Community Systems (I^2CS)

P-187 Paul Müller, Bernhard Neumair, Gabi Dreo Rodosek (Hrsg.)
4. DFN-Forum Kommunikations-technologien, Beiträge der Fachtagung 20. Juni bis 21. Juni 2011 Bonn

P-188 Holger Rohland, Andrea Kienle, Steffen Friedrich (Hrsg.)
DeLFI 2011 – Die 9. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. 5.–8. September 2011, Dresden

P-189 Thomas, Marco (Hrsg.)
Informatik in Bildung und Beruf
INFOS 2011
14. GI-Fachtagung Informatik und Schule

P-190 Markus Nüttgens, Oliver Thomas, Barbara Weber (Eds.)
Enterprise Modelling and Information Systems Architectures (EMISA 2011)

P-191 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2011
International Conference of the Biometrics Special Interest Group

P-192 Hans-Ulrich Heiß, Peter Pepper, Holger Schlingloff, Jörg Schneider (Hrsg.)
INFORMATIK 2011
Informatik schafft Communities

P-193 Wolfgang Lehner, Gunther Piller (Hrsg.)
IMDM 2011

P-194 M. Clasen, G. Fröhlich, H. Bernhardt, K. Hildebrand, B. Theuvsen (Hrsg.)
Informationstechnologie für eine nachhaltige Landbewirtschaftung
Fokus Forstwirtschaft

P-195 Neeraj Suri, Michael Waidner (Hrsg.)
Sicherheit 2012
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 6. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)

P-196 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2012
Proceedings of the 11th International Conference of the Biometrics Special Interest Group

P-197 Jörn von Lucke, Christian P. Geiger, Siegfried Kaiser, Erich Schweighofer, Maria A. Wimmer (Hrsg.)
Auf dem Weg zu einer offenen, smarten und vernetzten Verwaltungskultur
Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2012

P-198 Stefan Jähnichen, Axel Küpper, Sahin Albayrak (Hrsg.)
Software Engineering 2012
Fachtagung des GI-Fachbereichs Softwaretechnik

P-199 Stefan Jähnichen, Bernhard Rumpe, Holger Schlingloff (Hrsg.)
Software Engineering 2012
Workshopband

P-200 Gero Mühl, Jan Richling, Andreas Herkersdorf (Hrsg.)
ARCS 2012 Workshops

P-201 Elmar J. Sinz Andy Schürr (Hrsg.)
Modellierung 2012

P-202 Andrea Back, Markus Bick, Martin Breunig, Key Pousttchi, Frédéric Thiesse (Hrsg.)
MMS 2012:Mobile und Ubiquitäre Informationssysteme

P-203 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
5. DFN-Forum Kommunikations-technologien
Beiträge der Fachtagung

P-204 Gerald Eichler, Leendert W. M. Wienhofen, Anders Kofod-Petersen, Herwig Unger (Eds.)
12th International Conference on Innovative Internet Community Systems (I2CS 2012)

P-205 Manuel J. Kripp, Melanie Volkamer, Rüdiger Grimm (Eds.)
5th International Conference on Electronic Voting 2012 (EVOTE2012)
Co-organized by the Council of Europe, Gesellschaft für Informatik and E-Voting.CC

P-206 Stefanie Rinderle-Ma, Mathias Weske (Hrsg.)
EMISA 2012
Der Mensch im Zentrum der Modellierung

P-207 Jörg Desel, Jörg M. Haake, Christian Spannagel (Hrsg.)
DeLFI 2012: Die 10. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V.
24.–26. September 2012

P-208 Ursula Goltz, Marcus Magnor, Hans-Jürgen Appelrath, Herbert Matthies, Wolf-Tilo Balke, Lars Wolf (Hrsg.) INFORMATIK 2012

P-209 Hans Brandt-Pook, André Fleer, Thorsten Spitta, Malte Wattenberg (Hrsg.) Nachhaltiges Software Management

P-210 Erhard Plödereder, Peter Dencker, Herbert Klenk, Hubert B. Keller, Silke Spitzer (Hrsg.) Automotive – Safety & Security 2012 Sicherheit und Zuverlässigkeit für automobile Informationstechnik

P-211 M. Clasen, K. C. Kersebaum, A. Meyer-Aurich, B. Theuvsen (Hrsg.) Massendatenmanagement in der Agrar- und Ernährungswirtschaft Erhebung - Verarbeitung - Nutzung Referate der 33. GIL-Jahrestagung 20. – 21. Februar 2013, Potsdam

P-212 Arslan Brömme, Christoph Busch (Eds.) BIOSIG 2013 Proceedings of the 12th International Conference of the Biometrics Special Interest Group 04.–06. September 2013 Darmstadt, Germany

P-213 Stefan Kowalewski, Bernhard Rumpe (Hrsg.) Software Engineering 2013 Fachtagung des GI-Fachbereichs Softwaretechnik

P-214 Volker Markl, Gunter Saake, Kai-Uwe Sattler, Gregor Hackenbroich, Bernhard Mit schang, Theo Härder, Veit Köppen (Hrsg.) Datenbanksysteme für Business, Technologie und Web (BTW) 2013 13. – 15. März 2013, Magdeburg

P-215 Stefan Wagner, Horst Lichter (Hrsg.) Software Engineering 2013 Workshopband (inkl. Doktorandensymposium) 26. Februar – 1. März 2013, Aachen

P-216 Gunter Saake, Andreas Henrich, Wolfgang Lehner, Thomas Neumann, Veit Köppen (Hrsg.) Datenbanksysteme für Business, Technologie und Web (BTW) 2013 – Workshopband 11. – 12. März 2013, Magdeburg

P-217 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.) 6. DFN-Forum Kommunikations-technologien Beiträge der Fachtagung 03.–04. Juni 2013, Erlangen

P-218 Andreas Breiter, Christoph Rensing (Hrsg.) DeLFI 2013: Die 11 e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. (GI) 8. – 11. September 2013, Bremen

P-219 Norbert Breier, Peer Stechert, Thomas Wilke (Hrsg.) Informatik erweitert Horizonte INFOS 2013 15. GI-Fachtagung Informatik und Schule 26. – 28. September 2013

P-220 Matthias Horbach (Hrsg.) INFORMATIK 2013 Informatik angepasst an Mensch, Organisation und Umwelt 16. – 20. September 2013, Koblenz

P-221 Maria A. Wimmer, Marijn Janssen, Ann Macintosh, Hans Jochen Scholl, Efthimios Tambouris (Eds.) Electronic Government and Electronic Participation Joint Proceedings of Ongoing Research of IFIP EGOV and IFIP ePart 2013 16. – 19. September 2013, Koblenz

P-222 Reinhard Jung, Manfred Reichert (Eds.) Enterprise Modelling and Information Systems Architectures (EMISA 2013) St. Gallen, Switzerland September 5. – 6. 2013

P-223 Detlef Hühnlein, Heiko Roßnagel (Hrsg.) Open Identity Summit 2013 10. – 11. September 2013 Kloster Banz, Germany

P-224 Eckhart Hanser, Martin Mikusz, Masud Fazal-Baqaie (Hrsg.) Vorgehensmodelle 2013 Vorgehensmodelle – Anspruch und Wirklichkeit 20. Tagung der Fachgruppe Vorgehensmodelle im Fachgebiet Wirtschaftsinformatik (WI-VM) der Gesellschaft für Informatik e.V. Lörrach, 2013

P-225 Hans-Georg Fill, Dimitris Karagiannis, Ulrich Reimer (Hrsg.) Modellierung 2014 19. – 21. März 2014, Wien

P-226 M. Clasen, M. Hamer, S. Lehnert, B. Petersen, B. Theuvsen (Hrsg.) IT-Standards in der Agrar- und Ernährungswirtschaft Fokus: Risiko- und Krisenmanagement Referate der 34. GIL-Jahrestagung 24. – 25. Februar 2014, Bonn

P-227 Wilhelm Hasselbring,
Nils Christian Ehmke (Hrsg.)
Software Engineering 2014
Fachtagung des GI-Fachbereichs Softwaretechnik
25. – 28. Februar 2014
Kiel, Deutschland

P-228 Stefan Katzenbeisser, Volkmar Lotz, Edgar Weippl (Hrsg.)
Sicherheit 2014
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 7. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)
19.–21. März 2014, Wien

P-229 Dagmar Lück-Schneider, Thomas Gordon, Siegfried Kaiser, Jörn von Lucke,Erich Schweighofer, Maria A.Wimmer, Martin G. Löhe (Hrsg.)
Gemeinsam Electronic Government ziel(gruppen)gerecht gestalten und organisieren
Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2014, 20.-21. März 2014 in Berlin

P-230 Arslan Brömme, Christoph Busch (Eds.)
BIOSIG 2014
Proceedings of the 13th International Conference of the Biometrics Special Interest Group
10. – 12. September 2014 in Darmstadt, Germany

P-231 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
7. DFN-Forum Kommunikationstechnologien
16. – 17. Juni 2014
Fulda

P-232 E. Plödereder, L. Grunske, E. Schneider, D. Ull (Hrsg.)
INFORMATIK 2014
Big Data – Komplexität meistern
22. – 26. September 2014
Stuttgart

P-233 Stephan Trahasch, Rolf Plötzner, Gerhard Schneider, Claudia Gayer, Daniel Sassiat, Nicole Wöhrle (Hrsg.)
DeLFI 2014 – Die 12. e-Learning Fachtagung Informatik
der Gesellschaft für Informatik e.V.
15. – 17. September 2014
Freiburg

P-234 Fernand Feltz, Bela Mutschler, Benoît Otjacques (Eds.)
Enterprise Modelling and Information Systems Architectures
(EMISA 2014)
Luxembourg, September 25-26, 2014

P-235 Robert Giegerich, Ralf Hofestädt, Tim W. Nattkemper (Eds.)
German Conference on Bioinformatics 2014
September 28 – October 1
Bielefeld, Germany

P-236 Martin Engstler, Eckhart Hanser, Martin Mikusz, Georg Herzwurm (Hrsg.)
Projektmanagement und Vorgehensmodelle 2014
Soziale Aspekte und Standardisierung
Gemeinsame Tagung der Fachgruppen Projektmanagement (WI-PM) und Vorgehensmodelle (WI-VM) im Fachgebiet Wirtschaftsinformatik der Gesellschaft für Informatik e.V., Stuttgart 2014

P-237 Detlef Hühnlein, Heiko Roßnagel (Hrsg.)
Open Identity Summit 2014
4.–6. November 2014
Stuttgart, Germany

P-238 Arno Ruckelshausen, Hans-Peter Schwarz, Brigitte Theuvsen (Hrsg.)
Informatik in der Land-, Forst- und Ernährungswirtschaft
Referate der 35. GIL-Jahrestagung
23. – 24. Februar 2015, Geisenheim

P-239 Uwe Aßmann, Birgit Demuth, Thorsten Spitta, Georg Püschel, Ronny Kaiser (Hrsg.)
Software Engineering & Management 2015
17.-20. März 2015, Dresden

P-240 Herbert Klenk, Hubert B. Keller, Erhard Plödereder, Peter Dencker (Hrsg.)
Automotive – Safety & Security 2015
Sicherheit und Zuverlässigkeit für automobile Informationstechnik
21.–22. April 2015, Stuttgart

P-241 Thomas Seidl, Norbert Ritter, Harald Schöning, Kai-Uwe Sattler, Theo Härder, Steffen Friedrich, Wolfram Wingerath (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW 2015)
04. – 06. März 2015, Hamburg

P-242 Norbert Ritter, Andreas Henrich, Wolfgang Lehner, Andreas Thor, Steffen Friedrich, Wolfram Wingerath (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW 2015) – Workshopband
02. – 03. März 2015, Hamburg

P-243 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
8. DFN-Forum Kommunikationstechnologien
06.–09. Juni 2015, Lübeck

P-244 Alfred Zimmermann, Alexander Rossmann (Eds.)
Digital Enterprise Computing (DEC 2015)
Böblingen, Germany June 25-26, 2015

P-245 Arslan Brömme, Christoph Busch , Christian Rathgeb, Andreas Uhl (Eds.)
BIOSIG 2015
Proceedings of the 14th International Conference of the Biometrics Special Interest Group
09.–11. September 2015
Darmstadt, Germany

P-246 Douglas W. Cunningham, Petra Hofstedt, Klaus Meer, Ingo Schmitt (Hrsg.)
INFORMATIK 2015
28.9.-2.10. 2015, Cottbus

P-247 Hans Pongratz, Reinhard Keil (Hrsg.)
DeLFI 2015 – Die 13. E-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. (GI)
1.–4. September 2015
München

P-248 Jens Kolb, Henrik Leopold, Jan Mendling (Eds.)
Enterprise Modelling and Information Systems Architectures
Proceedings of the 6th Int. Workshop on Enterprise Modelling and Information Systems Architectures, Innsbruck, Austria
September 3-4, 2015

P-249 Jens Gallenbacher (Hrsg.)
Informatik allgemeinbildend begreifen
INFOS 2015 16. GI-Fachtagung Informatik und Schule
20.–23. September 2015

P-250 Martin Engstler, Masud Fazal-Baqaie, Eckhart Hanser, Martin Mikusz, Alexander Volland (Hrsg.)
Projektmanagement und Vorgehensmodelle 2015
Hybride Projektstrukturen erfolgreich umsetzen
Gemeinsame Tagung der Fachgruppen Projektmanagement (WI-PM) und Vorgehensmodelle (WI-VM) im Fachgebiet Wirtschaftsinformatik der Gesellschaft für Informatik e.V., Elmshorn 2015

P-251 Detlef Hühnlein, Heiko Roßnagel, Raik Kuhlisch, Jan Ziesing (Eds.)
Open Identity Summit 2015
10.–11. November 2015
Berlin, Germany

P-252 Jens Knoop, Uwe Zdun (Hrsg.)
Software Engineering 2016
Fachtagung des GI-Fachbereichs Softwaretechnik
23.–26. Februar 2016, Wien

P-253 A. Ruckelshausen, A. Meyer-Aurich, T. Rath, G. Recke, B. Theuvsen (Hrsg.)
Informatik in der Land-, Forst- und Ernährungswirtschaft
Fokus: Intelligente Systeme – Stand der Technik und neue Möglichkeiten
Referate der 36. GIL-Jahrestagung
22.-23. Februar 2016, Osnabrück

P-254 Andreas Oberweis, Ralf Reussner (Hrsg.)
Modellierung 2016
2.–4. März 2016, Karlsruhe

P-255 Stefanie Betz, Ulrich Reimer (Hrsg.)
Modellierung 2016 Workshopband
2.–4. März 2016, Karlsruhe

P-256 Michael Meier, Delphine Reinhardt, Steffen Wendzel (Hrsg.)
Sicherheit 2016
Sicherheit, Schutz und Zuverlässigkeit
Beiträge der 8. Jahrestagung des Fachbereichs Sicherheit der Gesellschaft für Informatik e.V. (GI)
5.–7. April 2016, Bonn

P-257 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
9. DFN-Forum Kommunikationstechnologien
31. Mai – 01. Juni 2016, Rostock

P-258 Dieter Hertweck, Christian Decker (Eds.)
Digital Enterprise Computing (DEC 2016)
14.–15. Juni 2016, Böblingen

P-259 Heinrich C. Mayr, Martin Pinzger (Hrsg.)
INFORMATIK 2016
26.–30. September 2016, Klagenfurt

P-260 Arslan Brömme, Christoph Busch, Christian Rathgeb, Andreas Uhl (Eds.)
BIOSIG 2016
Proceedings of the 15th International Conference of the Biometrics Special Interest Group
21.–23. September 2016, Darmstadt

P-261 Detlef Rätz, Michael Breidung, Dagmar Lück-Schneider, Siegfried Kaiser, Erich Schweighofer (Hrsg.)
Digitale Transformation: Methoden, Kompetenzen und Technologien für die Verwaltung
Gemeinsame Fachtagung Verwaltungsinformatik (FTVI) und Fachtagung Rechtsinformatik (FTRI) 2016
22.–23. September 2016, Dresden

P-262 Ulrike Lucke, Andreas Schwill, Raphael Zender (Hrsg.)
DeLFI 2016 – Die 14. E-Learning Fachtagung Informatik
der Gesellschaft für Informatik e.V. (GI)
11.–14. September 2016, Potsdam

P-263 Martin Engstler, Masud Fazal-Baqaie, Eckhart Hanser, Oliver Linssen, Martin Mikusz, Alexander Volland (Hrsg.)
Projektmanagement und Vorgehensmodelle 2016
Arbeiten in hybriden Projekten: Das Sowohl-als-auch von Stabilität und Dynamik
Gemeinsame Tagung der Fachgruppen Projektmanagement (WI-PM) und Vorgehensmodelle (WI-VM) im Fachgebiet Wirtschaftsinformatik
der Gesellschaft für Informatik e.V., Paderborn 2016

P-264 Detlef Hühnlein, Heiko Roßnagel, Christian H. Schunck, Maurizio Talamo (Eds.)
Open Identity Summit 2016
der Gesellschaft für Informatik e.V. (GI)
13.–14. October 2016, Rome, Italy

P-265 Bernhard Mitschang, Daniela Nicklas,Frank Leymann, Harald Schöning, Melanie Herschel, Jens Teubner, Theo Härder, Oliver Kopp, Matthias Wieland (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW 2017)
6.–10. März 2017, Stuttgart

P-266 Bernhard Mitschang, Norbert Ritter, Holger Schwarz, Meike Klettke, Andreas Thor, Oliver Kopp, Matthias Wieland (Hrsg.)
Datenbanksysteme für Business, Technologie und Web (BTW 2017)
Workshopband
6.–7. März 2017, Stuttgart

P-267 Jan Jürjens, Kurt Schneider (Hrsg.)
Software Engineering 2017
21.–24. Februar 2017, Hannover

P-268 A. Ruckelshausen, A. Meyer-Aurich, W. Lentz, B. Theuvsen (Hrsg.)
Informatik in der Land-, Forst- und Ernährungswirtschaft
Fokus: Digitale Transformation – Wege in eine zukunftsfähige Landwirtschaft
Referate der 37. GIL-Jahrestagung
06.–07. März 2017, Dresden

P-269 Peter Dencker, Herbert Klenk, Hubert Keller, Erhard Plödereder (Hrsg.)
Automotive – Safety & Security 2017
30.–31. Mai 2017, Stuttgart

P-270 Arslan Brömme, Christoph Busch, Antitza Dantcheva, Christian Rathgeb, Andreas Uhl (Eds.)
BIOSIG 2017
20.–22. September 2017, Darmstadt

P-271 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
10. DFN-Forum Kommunikations-technologien
30. – 31. Mai 2017, Berlin

P-272 Alexander Rossmann, Alfred Zimmermann (eds.)
Digital Enterprise Computing (DEC 2017)
11.–12. Juli 2017, Böblingen

P-273 Christoph Igel, Carsten Ullrich, Martin Wessner (Hrsg.)
BILDUNGSRÄUME
DeLFI 2017
Die 15. e-Learning Fachtagung Informatik der Gesellschaft für Informatik e.V. (GI)
5. bis 8. September 2017, Chemnitz

P-274 Ira Diethelm (Hrsg.)
Informatische Bildung zum Verstehen und Gestalten der digitalen Welt
13.–15. September 2017, Oldenburg

P-275 Maximilian Eibl, Martin Gaedke (Hrsg.)
INFORMATIK 2017
25.–29. September 2017, Chemnitz

P276 Alexander Volland, Martin Engstler, Masud Fazal-Baqaie, Eckhart Hanser, Oliver Linssen, Martin Mikusz (Hrsg.)
Projektmanagement und Vorgehensmodelle 2017
Die Spannung zwischen dem Prozess und den Menschen im Projekt
Gemeinsame Tagung der Fachgruppen Projektmanagement und Vorgehensmodelle im Fachgebiet Wirtschaftsinformatik der Gesellschaft für Informatik e.V. in Kooperation mit der Fachgruppe IT-Projektmanagement der GPM e.V., Darmstadt 2017

P-277 Lothar Fritsch, Heiko Roßnagel, Detlef Hühnlein (Hrsg.)
Open Identity Summit 2017
5.–6. October 2017, Karlstad, Sweden

P-278 Arno Ruckelshausen, Andreas Meyer-Aurich, Karsten Borchard, Constanze Hofacker, Jens-Peter Loy, Rolf Schwerdtfeger, Hans-Hennig Sundermeier, Helga Floto, Brigitte Theuvsen (Hrsg.)
Informatik in der Land-, Forst- und Ernährungswirtschaft
Referate der 38. GIL-Jahrestagung
26.–27. Februar 2018, Kiel

P-279 Matthias Tichy, Eric Bodden, Marco Kuhrmann, Stefan Wagner, Jan-Philipp Steghöfer (Hrsg.)
Software Engineering und Software Management 2018
5.–9. März 2018, Ulm

P-280 Ina Schaefer, Dimitris Karagiannis, Andreas Vogelsang, Daniel Méndez, Christoph Seidl (Hrsg.)
Modellierung 2018
21.–23. Februar 2018, Braunschweig

P-281 Hanno Langweg, Michael Meier, Bernhard C. Witt, Delphine Reinhardt (Hrsg.)
Sicherheit 2018
Sicherheit, Schutz und Zuverlässigkeit
25.–27. April 2018, Konstanz

P-282 Arslan Brömme, Christoph Busch, Antitza Dantcheva, Christian Rathgeb, Andreas Uhl (Eds.)
BIOSIG 2018
Proceedings of the 17th International Conference of the Biometrics Special Interest Group
26.–28. September 2018
Darmstadt, Germany

P-283 Paul Müller, Bernhard Neumair, Helmut Reiser, Gabi Dreo Rodosek (Hrsg.)
11. DFN-Forum Kommunikations-technologien
27.–28. Juni 2018, Günzburg

P-284 Detlef Krömker, Ulrik Schroeder (Hrsg.)
DeLFI 2018 – Die 16. E-Learning Fachtagung Informatik
10.–12. September 2018, Frankfurt a. M.

P-285 Christian Czarnecki, Carsten Brockmann, Eldar Sultanow, Agnes Koschmider, Annika Selzer (Hrsg.)
Workshops der INFORMATIK 2018 - Architekturen, Prozesse, Sicherheit und Nachhaltigkeit
26.–27. September 2018, Berlin

GI-Edition Lecture Notes in Informatics – Seminars

S-1 Johannes Magenheim, Sigrid Schubert (Eds.):
Informatics and Student Assessment
Concepts of Empirical Research and Standardisation of Measurement in the Area of Didactics of Informatics

S-2 Gesellschaft für Informatik (Hrsg.)
Informationstage 2005
Fachwissenschaftlicher Informatik-Kongress

S-3 Gesellschaft für Informatik (Hrsg.)
Informationstage 2006
Fachwissenschaftlicher Informatik-Kongress

S-4 Hans Hagen, Andreas Kerren, Peter Dannenmann (Eds.)
Visualization of Large and Unstructured Data Sets
First workshop of the DFG's International Research Training Group "Visualization of Large and Unstructured Data Sets – Applications in Geospatial Planning, Modeling and Engineering"

S-5 Gesellschaft für Informatik (Hrsg.)
Informationstage 2007
Fachwissenschaftlicher Informatik-Kongress

S-6 Gesellschaft für Informatik (Hrsg.)
Informationstage 2008
Fachwissenschaftlicher Informatik-Kongress

S-7 Hans Hagen, Martin Hering-Bertram, Christoph Garth (Eds.)
Visualization of Large and Unstructured Data Sets

S-8 Gesellschaft für Informatik (Hrsg.)
Informatiktage 2009
Fachwissenschaftlicher Informatik-Kongress

S-9 Gesellschaft für Informatik (Hrsg.)
Informatiktage 2010
Fachwissenschaftlicher Informatik-Kongress

S-10 Gesellschaft für Informatik (Hrsg.)
Informatiktage 2011
Fachwissenschaftlicher Informatik-Kongress

S-11 Gesellschaft für Informatik (Hrsg.)
Informatiktage 2012
Fachwissenschaftlicher Informatik-Kongress

S-12 Gesellschaft für Informatik (Hrsg.)
Informatiktage 2013
Fachwissenschaftlicher Informatik-Kongress

S-13 Gesellschaft für Informatik (Hrsg.)
Informatiktage 2014
Fachwissenschaftlicher Informatik-Kongress

S-14 Gesellschaft für Informatik (Hrsg.)
SKILL 2018
Studierendenkonferenz Informatik

The titles can be purchased at:

Köllen Druck + Verlag GmbH
Ernst-Robert-Curtius-Str. 14 · D-53117 Bonn
Fax: +49 (0)228/9898222
E-Mail: druckverlag@koellen.de

Gesellschaft für Informatik e.V. (GI)

publishes this series in order to make available to a broad public recent findings in informatics (i.e. computer science and information systems), to document conferences that are organized in cooperation with GI and to publish the annual GI Award dissertation.

Broken down into
• seminars
• proceedings
• dissertations
• thematics
current topics are dealt with from the vantage point of research and development, teaching and further training in theory and practice. The Editorial Committee uses an intensive review process in order to ensure high quality contributions.

The volumes are published in German or English.

Information: http://www.gi.de/service/publikationen/lni/

ISSN 1614-3213
ISBN 978-3-88579-448-6