International Journal on Advances in Internet Technology

2020 vol. 13 nr. 3&4

The *International Journal on Advances in Internet Technology* is published by IARIA.
ISSN: 1942-2652
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Sponsored by IARIA
www.iaria.org

International Journal on Advances in Internet Technology
Volume 13, Number 3 & 4, 2020

Tzung-Shi Chen, National University of Tainan, Taiwan
Xi Chen, University of Washington, USA
IlKwon Cho, National Information Society Agency, South Korea
Andrzej Chydzinski, Silesian University of Technology, Poland
Noël Crespi, Telecom SudParis, France
Antonio Cuadra-Sanchez, Indra, Spain
Javier Cubo, University of Malaga, Spain
Sagarmay Deb, Central Queensland University, Australia
Javier Del Ser, Tecnalia Research & Innovation, Spain
Philipe Devienne, LIFL - Université Lille 1 - CNRS, France
Kamil Dimililer, Near East Universiy, Cyprus
Martin Dobler, Vorarlberg University of Applied Sciences, Austria
Jean-Michel Dricot, Université Libre de Bruxelles, Belgium
Matthias Ehmann, Universität Bayreuth, Germany
Tarek El-Bawab, Jackson State University, USA
Nashwa Mamdouh El-Bendary, Arab Academy for Science, Technology, and Maritime Transport, Egypt
Mohamed Dafir El Kettani, ENSIAS - Université Mohammed V-Souissi, Morocco
Armando Ferro, University of the Basque Country (UPV/EHU), Spain
Anders Fongen, Norwegian Defence Research Establishment, Norway
Giancarlo Fortino, University of Calabria, Italy
Kary Främling, Aalto University, Finland
Steffen Fries, Siemens AG, Corporate Technology - Munich, Germany
Ivan Ganchev, University of Limerick, Ireland / University of Plovdiv “Paisii Hilendarski”, Bulgaria
Shang Gao, Zhongnan University of Economics and Law, China
Emiliano Garcia-Palacios, ECIT Institute at Queens University Belfast - Belfast, UK
Kamini Garg, University of Applied Sciences Southern Switzerland, Lugano, Switzerland
Rosario Giuseppe Garroppo, Dipartimento Ingegneria dell'informazione - Università di Pisa, Italy
Thierry Gayraud, LAAS-CNRS / Université de Toulouse / Université Paul Sabatier, France
Christos K. Georgiadis, University of Macedonia, Greece
Katja Gilly, Universidad Miguel Hernandez, Spain
Mariusz Głąbowski, Poznan University of Technology, Poland
Feliz Gouveia, Universidade Fernando Pessoa - Porto, Portugal
Kannan Govindan, Crash Avoidance Metrics Partnership (CAMP), USA
Bill Grosky, University of Michigan-Dearborn, USA
Jason Gu, Singapore University of Technology and Design, Singapore
Christophe Guéret, Vrije Universiteit Amsterdam, Nederlands
Frederic Guidec, IRISA-UBS, Université de Bretagne-Sud, France
Bin Guo, Northwestern Polytechnical University, China
Gerhard Hancke, Royal Holloway / University of London, UK
Arthur Herzog, Technische Universität Darmstadt, Germany
Rattikorn Hewett, Whitacre College of Engineering, Texas Tech University, USA
Quang Hieu Vu, EBTIC, Khalifa University, Arab Emirates
Hiroaki Higaki, Tokyo Denki University, Japan
Dong Ho Cho, Korea Advanced Institute of Science and Technology (KAIST), Korea
Anna Hristoskova, Ghent University - IBBT, Belgium
Ching-Hsien (Robert) Hsu, Chung Hua University, Taiwan
Chi Hung, Tsinghua University, China
Edward Hung, Hong Kong Polytechnic University, Hong Kong
Raj Jain, Washington University in St. Louis , USA
Edward Jaser, Princess Sumaya University for Technology - Amman, Jordan
Terje Jensen, Telenor Group Industrial Development / Norwegian University of Science and Technology, Norway
Yasushi Kambayashi, Nippon Institute of Technology, Japan
Georgios Kambourakis, University of the Aegean, Greece

Atsushi Kanai, Hosei University, Japan
Henrik Karstoft , Aarhus University, Denmark
Dimitrios Katsaros, University of Thessaly, Greece
Ayad ali Keshlaf, Newcastle University, UK
Reinhard Klemm, Avaya Labs Research, USA
Samad Kolahi, Unitec Institute Of Technology, New Zealand
Dmitry Korzun, Petrozavodsk State University, Russia / Aalto University, Finland
Slawomir Kuklinski, Warsaw University of Technology, Poland
Andrew Kusiak, The University of Iowa, USA
Mikel Larrea, University of the Basque Country UPV/EHU, Spain
Frédéric Le Mouël, University of Lyon, INSA Lyon / INRIA, France
Juong-Sik Lee, Nokia Research Center, USA
Wolfgang Leister, Norsk Regnesentral (Norwegian Computing Center), Norway
Clement Leung, Hong Kong Baptist University, Hong Kong
Longzhuang Li, Texas A&M University-Corpus Christi, USA
Yaohang Li, Old Dominion University, USA
Jong Chern Lim, University College Dublin, Ireland
Lu Liu, University of Derby, UK
Damon Shing-Min Liu, National Chung Cheng University, Taiwan
Michael D. Logothetis, University of Patras, Greece
Malamati Louta, University of Western Macedonia, Greece
Maode Ma, Nanyang Technological University, Singapore
Elsa María Macías López, University of Las Palmas de Gran Canaria, Spain
Olaf Maennel, Loughborough University, UK
Zoubir Mammeri, IRIT - Paul Sabatier University - Toulouse, France
Yong Man, KAIST (Korea advanced Institute of Science and Technology), South Korea
Sathiamoorthy Manoharan, University of Auckland, New Zealand
Chengying Mao, Jiangxi University of Finance and Economics, China
Brandeis H. Marshall, Purdue University, USA
Constandinos Mavromoustakis, University of Nicosia, Cyprus
Shawn McKee, University of Michigan, USA
Stephanie Meerkamm, Siemens AG in Erlangen, Germany
Kalogiannakis Michail, University of Crete, Greece
Peter Mikulecky, University of Hradec Kralove, Czech Republic
Moeiz Miraoui, Université du Québec/École de Technologie Supérieure - Montréal, Canada
Shahab Mokarizadeh, Royal Institute of Technology (KTH) - Stockholm, Sweden
Mario Montagud Climent, Polytechnic University of Valencia (UPV), Spain
Stefano Montanelli, Università degli Studi di Milano, Italy
Julius Müller, TU- Berlin, Germany
Juan Pedro Muñoz-Gea, Universidad Politécnica de Cartagena, Spain
Krishna Murthy, Global IT Solutions at Quintiles - Raleigh, USA
Alex Ng, University of Ballarat, Australia
Christopher Nguyen, Intel Corp, USA
Petros Nicopolitidis, Aristotle University of Thessaloniki, Greece
Carlo Nocentini, Università degli Studi di Firenze, Italy
Federica Paganelli, CNIT - Unit of Research at the University of Florence, Italy
Carlos E. Palau, Universidad Politecnica de Valencia, Spain
Matteo Palmonari, University of Milan-Bicocca, Italy
Ignazio Passero, University of Salerno, Italy
Serena Pastore, INAF - Astronomical Observatory of Padova, Italy
Fredrik Paulsson, Umeå University, Sweden
Rubem Pereira, Liverpool John Moores University, UK
Yulia Ponomarchuk, Far Eastern State Transport University, Russia

Jari Porras, Lappeenranta University of Technology, Finland
Neeli R. Prasad, Aalborg University, Denmark
Drogkaris Prokopios, University of the Aegean, Greece
Emanuel Puschita, Technical University of Cluj-Napoca, Romania
Lucia Rapanotti, The Open University, UK
Gianluca Reali, Università degli Studi di Perugia, Italy
Jelena Revzina, Transport and Telecommunication Institute, Latvia
Karim Mohammed Rezaul, Glyndwr University, UK
Leon Reznik, Rochester Institute of Technology, USA
Simon Pietro Romano, University of Napoli Federico II, Italy
Michele Ruta, Technical University of Bari, Italy
Jorge Sá Silva, University of Coimbra, Portugal
Sébastien Salva, University of Auvergne, France
Ahmad Tajuddin Samsudin, Telekom Malaysia Research & Development, Malaysia
Josemaria Malgosa Sanahuja, Polytechnic University of Cartagena, Spain
Luis Enrique Sánchez Crespo, Sicaman Nuevas Tecnologías / University of Castilla-La Mancha, Spain
Paul Sant, University of Bedfordshire, UK
Brahmananda Sapkota, University of Twente, The Netherlands
Alberto Schaeffer-Filho, Lancaster University, UK
Peter Schartner, Klagenfurt University, System Security Group, Austria
Rainer Schmidt, Aalen University, Germany
Thomas C. Schmidt, HAW Hamburg, Germany
Zary Segall, Chair Professor, Royal Institute of Technology, Sweden
Dimitrios Serpanos, University of Patras and ISI/RC ATHENA, Greece
Jawwad A. Shamsi, FAST-National University of Computer and Emerging Sciences, Karachi, Pakistan
Michael Sheng, The University of Adelaide, Australia
Kazuhiko Shibuya, The Institute of Statistical Mathematics, Japan
Roman Y. Shtykh, Rakuten, Inc., Japan
Patrick Siarry, Université Paris 12 (LiSSi), France
Jose-Luis Sierra-Rodriguez, Complutense University of Madrid, Spain
Simone Silvestri, Sapienza University of Rome, Italy
Vasco N. G. J. Soares, Instituto de Telecomunicações / University of Beira Interior / Polytechnic Institute of Castelo Branco, Portugal
Radosveta Sokullu, Ege University, Turkey
José Soler, Technical University of Denmark, Denmark
Victor J. Sosa-Sosa, CINVESTAV-Tamaulipas, Mexico
Dora Souliou, National Technical University of Athens, Greece
João Paulo Sousa, Instituto Politécnico de Bragança, Portugal
Kostas Stamos, Computer Technology Institute & Press "Diophantus" / Technological Educational Institute of Patras, Greece
Cristian Stanciu, University Politehnica of Bucharest, Romania
Vladimir Stantchev, SRH University Berlin, Germany
Tim Strayer, Raytheon BBN Technologies, USA
Masashi Sugano, School of Knowledge and Information Systems, Osaka Prefecture University, Japan
Tae-Eung Sung, Korea Institute of Science and Technology Information (KISTI), Korea
Sayed Gholam Hassan Tabatabaei, Isfahan University of Technology, Iran
Yutaka Takahashi, Kyoto University, Japan
Yoshiaki Taniguchi, Kindai University, Japan
Nazif Cihan Tas, Siemens Corporation, Corporate Research and Technology, USA
Alessandro Testa, University of Naples "Federico II" / Institute of High Performance Computing and Networking (ICAR) of National Research Council (CNR), Italy
Stephanie Teufel, University of Fribourg, Switzerland
Parimala Thulasiraman, University of Manitoba, Canada

Pierre Tiako, Langston University, USA
Orazio Tomarchio, Universita' di Catania, Italy
Dominique Vaufreydaz, INRIA and Pierre Mendès-France University, France
Krzysztof Walkowiak, Wroclaw University of Technology, Poland
MingXue Wang, Ericsson Ireland Research Lab, Ireland
Wenjing Wang, Blue Coat Systems, Inc., USA
Zhi-Hui Wang, School of Softeware, Dalian University of Technology, China
Matthias Wieland, Universität Stuttgart, Institute of Architecture of Application Systems (IAAS),Germany
Bernd E. Wolfinger, University of Hamburg, Germany
Chai Kiat Yeo, Nanyang Technological University, Singapore
Abdulrahman Yarali, Murray State University, USA
Mehmet Erkan Yüksel, Istanbul University, Turkey

International Journal on Advances in Internet Technology
Volume 13, Numbers 3 & 4, 2020

CONTENTS

Measuring Competence: Improvements to Determine the Degree of Opinion Leadership in Social Networks

Michael Spranger*†, Kai-Jannis Hanke†, Florian Heinke† and Dirk Labudde†‡

†University of Applied Sciences Mittweida
Forensic Science Investigation Lab (FoSIL), Germany
Email: *name.surname*@hs-mittweida.de
‡Fraunhofer
Cyber Security
Darmstadt, Germany
Email: labudde@hs-mittweida.de

***Abstract*—In recent years, the automated, efficient and sensitive monitoring of social networks has become increasingly important for the criminal investigation process and crime prevention. Previously, we have shown that the detection of opinion leaders is of great interest in forensic applications to gather important information. In the current work, it is argued that state of the art methods, determining the relative degree to which an opinion leader exerts influence over the network, have weaknesses if networks exhibit a star-like social graph topology, whereas these topologies result from the interaction of users with similar interests. This is typically the case in networks of political organizations. In these cases, the underlying topologies are highly focused on one (or only a few) central actor(s) and lead to less meaningful results by classic measures of node centrality commonly used to ascertain the degree of leadership. With the help of data collected from the Facebook and Twitter network of a German political party, these aspects are examined and a quantitative indicator for describing star-like network topologies is introduced and discussed. This measure can be of great value in assessing the applicability of established leader detection methods. Finally, two variations of a new measure– the CompetenceRank – which is based on the LeaderRAnk score and aims to address the discussed problems in cases with and without additional network data such as *likes* and *shares*, are proposed.**

Keywords–Forensic; Opinion Leader; Graph Theory.

I. Introduction

The detection of opinion leaders in online social networks has been discussed extensively over the past few years. While the term "detection" is generally associated with a binary decision, here – in accordance with other papers in this domain – it is used to refer to the determination of the degree of leadership. The scope of application is manifold and reaches from determining influencers and brand ambassadors up to finding those who influence the political opinion of a group of people. Especially the last application can be of interest to law enforcement and intelligence agencies. In [1] it was shown that in some situations previous approaches based on the work by Katz [2], who focused on networks in the offline world, do not capture the core of the problem and as a result lead to an inaccurate assessment of opinion leadership.

Measures for opinion leadership on social networks tend to focus on a single aspect: network contribution. However, it becomes clear that only evaluating network contribution such as posting content, commenting it or replying to it does not capture the full range of interactions social media platforms have to offer. Besides network contribution or content generation in the ordinary sense we also find a secondary form of participation, which solely relies on existing content. Virtually nodding in agreement by clicking *like* or extending the reach of a given post by sharing it, is not creating new content in a given network. However, measures reflecting such activities exist on most social media platforms and play a substantial role in determining ones reach and authority. These secondary measures do not only shape how people interact but also influence who rises to the position of an opinion leader.

This section shall give a brief introduction to the field in which situations may occur, in which the LeaderRank leads to inappropriate results. Furthermore, it will give an overview of topology-based approaches and it finishes with the scope and structure of the paper.

A. General Motivation

Analyzing social networks has become an important tool for investigators, intelligence services and decision makers of police services. The information gained this way can be used to solve crimes by searching for digital evidence that relates to the crime in the real world. Additionally, methods of predictive policing can help to organize police missions as was shown in [3]–[5]. The detection of opinion leaders in social networks is an important task for different reasons. On the one hand, owners of influential profiles are often also influential in the offline world. Knowing these people helps to determine the direction of an investigation or more concretely to target persons of interest. On the other hand, as was suggested in previous work [5], it might be of interest to contact these profiles by means of chatbots to gain access into closed groups in an effort to gather important information for intelligence services. Intuitively, opinion leaders, when considered as nodes with high structural importance, can be detected with the help of centrality measures. However, different kinds of influence in a network have to be distinguished. Nodes can have a great influence as corresponding actors are able to spread information fast and widely in a network, or they can have a great influence because they write something of importance that attracts many other users in the network to respond.

B. Leader Detection by means of Network Centrality Measures

In the literature, one can mainly find centrality measures for the former type of influence. For example, highly active profiles can be recognized using degree centrality, meaning, the relative number of outgoing edges of a node. These profiles are represented by nodes with a high degree centrality and are especially useful to spread information in a network due to their high interconnectedness. In this context, the closeness centrality – the inverse of the mean of the shortest path of a node to any other node in the network – is even more effective. It describes the efficiency of the dissemination of information of a certain node.

Furthermore, the betweenness centrality of a certain node, which is defined as the number of shortest paths between two nodes that cross this node, describes the importance of this node for the dissemination of information in a network. Therefore, the higher the betweenness centrality of a node, the greater its importance for the exchange of information in a network.

Moreover, the eigenvector centrality of a node is defined as the principal eigenvector of the adjacency matrix of a network. In contrast to the measures discussed beforehand, PageRank [6], as one of the best measures of node centrality, does not only consider the centrality of the node itself, yet also of its neighboring nodes.

As part of the opinion leader detection research, LeaderRank [7] was introduced as a further development of PageRank in order to find nodes that spread information further and faster. However, all of these centrality measures consider nodes that are involved in the dissemination of information mainly based on their activity. For the purpose of the intended usage, users who achieve high impact through what they have written are of much greater interest. Thus, similar to the citation of papers and books and its impact on the author's reputation, the importance of a node has to be higher when it reaches a high number of references and citations with low activity.

Especially social media platforms provide comparable metrics, such as *likes* and *shares* that partially reflect the author's reputation and credibility. Hence, it is imperative to consider respective measures of acceptance, expertise and authority when determining opinion leaders in any digital social network.

Interestingly, Li et al. considered the so-called node spreadability as the ground truth for quantifying node importance in a subsequent study [8]. Subsequently, node spreadability is based on a straightforward Susceptible-Infected-Removed (SIR) infection model from which the expected number of infected nodes upon initially infecting the node in question is estimated. However, this expected number can only be estimated from simulation, which, furthermore, is dependent on the parameterization of the SIR model. In this respect, all centrality measures can be considered as heuristic approximations of node spreadability.

C. Scope and Structure of the Paper

In this work, we discuss problems that can arise when aiming to detect opinion leaders in social networks yielding highly central topologies similar to star graphs. Examples for such networks are especially group pages on Facebook or vk.com where user interactions and activities are mostly triggered by and focused on posts made by the page owner. In such cases, the page owner – a trivial leader in the sense of centrality measures discussed above – acts as a score aggregator and can thus lead to distorted scoring, which can eventually be adverse in the context of opinion leader detection. In this case, classic centrality measures can be considered inappropriate. Based on interactions of users of the Facebook page of the German political pary "DIE LINKE" tracked for five consecutive months (January - May 2017), this problem is illustrated. We further introduce the LeaderRank skewness as a quantitative measure of aggregator-induced distorted LeaderRank scoring, which in experiments show to be superior to network entropy with respect to expressiveness. Additionally, a simple modified LeaderRank score, to which we refer to as CompetenceRank, is introduced. It is proposed to be more suitable for opinion leader detection in such networks, especially, if additional data for *likes* and *shares* are not available.

For such cases in which these data is available an improved version of the CompetenceRank is proposed and evaluated using the Twitter network of "DIE LINKE". The corresponding data set contains not only tweets, comments and replies from the entire year 2018, it also incorporates the accompanying *like* and *retweet* counts for each tweet, comment and reply. In politically motivated networks, as the one analyzed in this paper, the improved CompetenceRank shows a substantial increase in performance compared to the LeaderRank and the simple CompetenceRank.

The paper is structured as follows: in Section II, a brief literature overview on the topic of opinion leader detection is given, followed by a summary of the LeaderRank algorithm. In Section III two shortcomings of the LeaderRank are discussed: firstly, the skewness of the rank distribution in star-shaped network topologies and, secondly, that not all available data of social media platforms are taken into account. Subsequently, in the same section the deduction and definition of the normalized LeaderRank skewness as a metric for an approximation of a star-shaped topology is discussed and compared with the normalized graph entropy. In Section IV three datasets are introduced, which were used to evaluate these metrics, two of which were also used to develop solutions for the aforementioned problems as proposed in Section V by introducing the CompetenceRank for taking authority into account as well as an improvement for cases in which additional data is available. Subsequently, Section VI contains an evaluation of both CompetenceRank versions using the Twitter network. Finally, a conclusion as well as an overview of future work is given in Section VII.

II. Detection of Opinion Leaders

Opinion leaders in the context of the intended analysis of social networks are individuals, who exert a significant amount of influence on the opinion and sentiment of other users of the network through their actions or by what they are communicating. In social sciences the term "opinion leader" was introduced before 1957 by Katz and Lazarsfeld's research on diffusion theory [2]. Their proposed two-step flow model retains validity in the digital age, especially in the context of social media.

Katz et al. assume that information disseminated in a social network is received, strengthened and enriched by opinion

leaders in their social environment. Each individual is influenced in his opinion by a variety of heterogeneous opinion leaders. This signifies that the opinion of an individual is mostly formed by its social environment. In 1962, Rogers referenced these ideas and defined opinion leader as follows:

> "Opinion leadership is the degree to which an individual is able to influence informally other individuals' attitudes or overt behavior in a desired way with relative frequency." [9, p. 331]

For the present study, one important question to answer is what influence means, or rather how to identify an opinion leader or how the influencer can be distinguished from those being influenced. Katz defined the following features [2]:

1) personification of certain values,
2) competence,
3) strategic social location.

One approach to identify opinion-leaders is to extract and analyze the content of nodes and edges of networks to mine leadership features. For instance, the sentiment of communication pieces can be analyzed to detect the influence of their authors, as shown by Huang et. al., who aim to detect the most influential comments in a network this way [10]. Another strategy is to perform topic mining to categorize content and detect opinion leaders for each topic individually, as opinion leadership is context-dependent [2] [11]. For this purpose, Latent Dirichlet Allocation (LDA) [12] can be used, as seen in the work of [13]. Furthermore, Aleahmad et. al. achieved good results with OLFinder by utilizing both topic mining methods and centrality measures [14]. Additionally, Chen et al. proposed D_OLMiner, which derives opinion leaders from dynamic social networks [15].

Another novel approach, the firefly algorithm, a meta-heuristic optimization algorithm that can deal with especially large networks, is based on the behavior of fireflies and is used by Jain et. al. to determine local and global opinion leaders [16].

For this study, we considered the implementation of content-based methods problematic, as texts in social networks mostly lack correct spelling and formal structure which impairs such methods' performance. Additionally, leaders can be identified by analyzing the flow of information in a network. By monitoring how the interaction of actors evolves over time, one can identify patterns and individuals of significance within them. To achieve this, some model of information propagation is required, such as Markov processes employed by [17] and the probabilistic models proposed by [18]. These interaction-based methods consider both topological features and their dynamics over time. DDOL is a recent, dynamic approach by Queslati et. al. that focuses on social signals (shares, comments, likes) and terms that are frequently encountered in the expression of opinions. DDOL does not include centrality measures and has a slightly lower precision than PageRank but contrary to PageRank it works on dynamic networks and a has a lower computational complexity [19].

Parts of this study use methods that are solely based on a network's topology, therefore, considering features, such as node degree, neighborhood distances and clusters, to identify opinion leaders. One implementation for the former is the calculation of node centrality. The underlying assumption is that the more influence an individual gains, the more central it is in the network. Which centrality measure is most suitable is dependent on the application domain. We judged eigenvector centrality to be most adequate. One of the most popular algorithms is Google's PageRank algorithm [6]. The application of PageRank for the purposes of opinion leader detection has seen merely moderate success [20] [21].

With LeaderRank scores, Lü et al. advocate further development and optimization of this algorithm for social networks, and have achieved surprisingly good results [7]. Herein, users are considered as vertices and directed edges as relationships between opinion leaders and users. All users are also bidirectionally connected to a ground vertex, which ensures connectivity as well as score convergence. In short, the algorithm is an iterative multiplication of a vector comprised by per-vertex scores $s_i(t)$ at iteration step t with a weighted adjacency matrix until convergence is achieved according to some convergence criteria. Initially, at iteration step t_0, all vertex scores are set to $s(0) = 1$, except for the ground vertex score which is initialized as $s_g(0) = 0$. Equation (1) describes the LeaderRank algorithm as a model of probability flow through the network, where $s_i(t)$ indicates the score of a vertex i at iteration step t.

$$s_i(t+1) = \sum_{j=1}^{N+1} \frac{a_{ji}}{e_{v_j}^{out}} s_j(t) \tag{1}$$

Depending on whether or not there exists a directed edge from vertex j to the vertex i, the value 1 respectively 0 is assigned to a_{ji}. $e_{v_j}^{out}$ describes the number of outgoing edges of a vertex j. The update rule given in Equation (1) can be rewritten as a matrix-vector product:

$$\mathbf{s}(t+1) = \tilde{\mathbf{A}}\mathbf{s}(t), \tag{2}$$

where $\mathbf{s}(t)$ corresponds to the vector of the $N+1$ vertex scores at iteration step t, and $\tilde{\mathbf{A}}$ is the weighted adjacency matrix of size $(N+1) \times (N+1)$ with

$$\tilde{A}_{ji} = \frac{a_{ji}}{e_{v_j}^{out}}. \tag{3}$$

The final score is obtained as the score of the respective vertex at the convergence step t_c and the obtained ground vertex score, as shown in (4). At t_c, equilibration of LeaderRank scores towards a steady state is observed.

$$\mathrm{S}_i = s_i(t_c) + \frac{s_g(t_c)}{N} \tag{4}$$

Furthermore, note that

$$\sum_{i=1}^{N} S_i = \sum_{i=1}^{N} s_i(t) = N. \tag{5}$$

The advantage of this algorithm compared to PageRank is that the convergence is faster and, above all, that vertices that spread information faster and further can be found. In later work, for example, by introducing a weighting factor, as in [8] or [22], susceptibility to noisy data has been further reduced and the ability to find influential distributors (hubs) of information has been added.

III. Issues with LeaderRank

The LeaderRank algorithm can be understood as a reversion of a discrete model of diffusion. In that sense, the initialization $s_i(0) = 1$ at t_0 can be interpreted as assigning a uniform concentration distribution of some virtual compound that, in the processes, is re-distributed according to the model. In that respect, central actors showing the highest activity in star-like networks can induce score aggregation and migration towards their central nodes as well as their adjacent nodes, whereas nodes in the 'peripheral region' of the network become inadequately represented by their scores. Therefore, one can hypothesize that ranked lists obtained from LeaderRank scores can not be considered meaningful if a given network in question exhibits a star-like topology.

Another problem of LeaderRank comes into existence when considering means of communication that differ from traditional ones in person dialogues. Most social media platforms utilize *likes*, *shares*, *dislikes* and the concept of building a follower base. The amount of, for example, *likes* that a post receives or the frequency with which it is shared indicate its importance within a network and at least partially reflect the influence of the respective author. In turn, such data should be included when determining opinion leadership. Theoretically, LeaderRank has the capacity to incorporate aforementioned additional data. However, if this data were to be included in a network graph, then each *like*, *share* or anything similar would be seen as a unique edge from one node to another, just like regular forms of communication. This introduces two major problems, a theoretical one and a practical one. Firstly, is a *like* on a post equally as valuable as an actual reply and then how influential is a *share*? Evidently, there is a difference between the interaction activities, such as liking, sharing, writing or replying to a post, but this discrepancy is difficult to capture with the LeaderRank. Either one accepts that *likes* and *shares* have similar value to a written reply or one needs to additionally implement weights for different types of edges within a network.

Secondly, including *likes* as edges between nodes poses a practical problem: partial networks. When considering an individual post, then ideally the name of every individual who has liked this post is available in our data set, but in a real world example this is usually not the case. For example, when analyzing a twitter network one can discover how many people liked an individual post quite easily, but recovering the names of those individuals is highly restricted as twitter only provides a shortened list of names. It might be possible to recover all the names for a tweet with only 15 *likes*, but the list of names for a tweet with 100 *likes* can have the same length as the list for a tweet with 1.000 *likes*. Clearly, we lose a significant amount of information with exactly those tweets that are of great interest for opinion leadership, that is, tweets with seemingly the most influence over other users. When faced with similar restrictions on different platforms the total count of *likes* or *shares* might be more useful than a drastically reduced and limited list of names. In a similar manner it makes more sense to determine the popularity of politicians by counting the attendees of a political event compared to getting the names of only the first hundred attendees. Hence, it makes more sense to define people posting on social media as "politicians" speaking on a stage whereas users liking or sharing their content can be seen as attendees nodding in agreement or sending pictures of the stage to their friends.

On social media we have many attendees, virtually nodding their heads by clicking *like* or retweeting or sharing interesting content but they do not contribute by producing new posts. Incomplete data sets may not include the name for every person that likes a contribution, but these users can still be influenced and may even shape the network, since *likes* and *shares* present a measure for authority, credibility and approval in a given network. As a result, accounts partaking in the network through *likes* and *shares* should receive recognition as they silently enable cognitive biases, like the bandwagon effect [23] or herding mentality [24], that in turn alter how well-liked content appears to be, consequently, making it more or less influential. Ideally, LeaderRank does not only find opinion leaders in complete networks, but also discovers them in incomplete data sets. As a result, accounts that cannot be represented in the graph due to the absence of a name should still be considered when determining opinion leadership. A magnitude of nameless accounts cannot be included in a graph and thus they will not receive LeaderRank-Scores themselves, but seen as a collective they may help in shaping a network and identifying truly influential opinion leaders.

In this case study, two different networks are being examined. Namely, the network around the Facebook page as well as the Twitter network of the German left-winged political party "DIE LINKE". Firstly, the star topology of the Facebook network is being evaluated and secondly a novel approach to include *likes* and *retweets* is tested on the Twitter network.

In the first case study, the Facebook network under investigation shows an extreme case of a star topology in which the owner of the political Facebook page "DIE LINKE" acts solely as the central actor (for more information see Section IV). Since the LeaderRank emphasizes the strategic social location of a user, their competence seems to be improperly valued. In star-shaped network topologies, high centralities of only a fraction of nodes leads to a heavily skewed LeaderRank score distribution.

In contrast, one could argue that someone is more important if any activity generates a high number of responses. Such a case is regularly given by political networks which are dominated by the central node of the page owner. Consequently, a straightforward modification of the LeaderRank score is proposed in Section V-A addressing the imbalance the LeaderRank algorithm yields in such networks.

In the following paragraph a quantitative measure of LeaderRank distribution skewness is proposed that could aid to ensure proper applicability of the LeaderRank algorithm for any given network. This measure is further compared to the classic measure of network entropy. Tests on simulated data show the LeaderRank skewness to be superior to network entropy with respect to topological changes.

A. Definition of LeaderRank Distribution Skewness

Let $LR = \{S_1, ..., S_i, ..., S_N\}$ be the LeaderRank scores of all nodes. Further, $\overline{S}$ and sd_{LR} denote the arithmetic mean and standard deviation of LR. Based on the z-scaled LeaderRank scores (6), the skewness ν of the LeaderRank distribution is calculated as shown in (7).

$$z(S_i) = \frac{S_i - \overline{S}}{sd_{LR}} \tag{6}$$

$$\nu_{LR} = \left| \frac{1}{N} \sum_i z(S_i)^3 \right| \quad (7)$$

As discussed above, score distribution skewness is correlated with network topology. Yet, normalization of computed skewness is required in order to make a statement about the topology and whether a star-like topology is present. Hence, upper and lower bounds, ν_{min} and ν_{max}, are needed. In this paragraph, derivation of both bounds are given.

Trivially, ν converges to the lower bound – the theoretical minimum ($\nu = 0$) – in almost-regular graphs. Such graphs are regular graphs with one edge being removed. With N being sufficiently large, the supposition that $S_i \approx S_j$ for any pair of randomly selected vertices of a social network graph $v_i, v_j \in V$ holds true and a limit of $\lim_{sd_{LR} \to 0} \nu = 0$ can be assumed. In regular graphs however all LeaderRank scores are equal by definition, resulting to $sd_{LR} = 0$ and ν being undefined in this case.

In contrast, ν is equal to the theoretical maximum if the network graph exhibits a strictly star-shaped topology. Directed star graphs are graphs with a central vertex v_c and $N-1$ leaf vertices connected to v_c. One can re-write the set of star graph vertices as $V = \{v_c, v_2, ..., v_N\}$ and denote the LeaderRank score set as $LR = \{S_c, S_2, ..., S_N\}$.The LeaderRank scores of any randomly selected pair of vertices v_i and v_j with $v_i, v_j \neq v_c$, with v_c being the central vertex, are then not distinguishable, i. e., $S_i = S_j$, according to the LeaderRank's definition. Furthermore, the sum of LeaderRank scores equals N leading to $\overline{S} = 1$ for any given graph. Given the central node's score S_c, each S_i can thus be calculated as shown in (8).

$$S_i = \frac{N - S_c}{N - 1} \quad (8)$$

Thus if S_c is known, the set of LeaderRank values $\{S_c, S_2, ...S_i, ..., S_N\}$ and the resulting ν_{max} can be derived. In the following text we shall give an explicit relationship between the number of nodes N in a directed star graph and the corresponding score set LR. For this, let $\mathbf{s}$ be the scores vector at the steady-state to which $\mathbf{s}(t)$ converges according to the update rule (see Equation (2)). Then the identity given in Equation (9) holds, since $\mathbf{s} = \mathbf{s}(t+1) = \mathbf{s}(t)$.

$$\mathbf{s} = \tilde{\mathbf{A}}\mathbf{s} \quad (9)$$

Thus equation (9), in conjunction with the relation given in equation (5), yields a set of $N+2$ equations from which $\mathbf{s}$ can be (theoretically) obtained for any given graph, if a sufficiently efficient solver algorithm exists. However, for directed star graphs solving these equations is straight-forward, and leads to an explicit formalism for $\mathbf{s}$ and the LeaderRank scores LR accordingly. Solving this set of equations involves that $\tilde{\mathbf{A}}$ can be explicitly written as

$$\tilde{\mathbf{A}} = \begin{pmatrix} 0 & 1/2 & 1/2 & ... & 1/2 & 1/N \\ 0 & 0 & 0 & ... & 0 & 1/N \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & ... & 0 & 1/N \\ 1 & 1/2 & 1/2 & ... & 1/2 & 1/N \end{pmatrix}. \quad (10)$$

for any given directed, extended star graph with vertices $V = \{v_c, v_2, ..., v_N, v_g\}$. One henceforth obtains the steady-state score vector $\mathbf{s} = (s_c, s_2, ..., s_N, s_g)^\intercal$ from the resulting set of equations which can be derived by simply re-arranging Equations (9) and (5):

$$s_c = \frac{N^2}{5N-1} + \frac{N}{5N-1} \quad (11)$$

$$s_i = \frac{2N}{5N-1}, \forall i = 2, ..., N \quad (12)$$

$$s_g = \frac{2N^2}{5N-1}. \quad (13)$$

This explicit formalism of $\tilde{\mathbf{A}}$ also highlights that the leaf vertices (denoted as v_i for textual cleanness in the following text) are indistinguishable with respect to the weighted adjacency matrix values $\tilde{A}_{i\cdot}$. Thus, the obtained LeaderRank scores S_i are identical as well. Plugging the computed values of $\mathbf{s}$ into the final update rule (see Equation (4)) yields the LeaderRank score for the central vertex v_c:

$$S_c = \frac{N^2}{5N-1} + \frac{3N}{5N-1} \quad (14)$$

$$(15)$$

Then the equal LeaderRank score S_i of the leaf nodes can be calculated according to Equation (8), from which the upper skewness bound ν_{max} is readily computed. Subsequently, for any irregular network graph the LeaderRank skewness can be calculated and normalized subsequently using a min-max normalization as denoted in (16), whereas ν_{min} can be assumed as 0 as discussed above.

$$\hat{\nu} = \frac{\nu - \nu_{min}}{\nu_{max} - \nu_{min}} = \frac{\nu}{\nu_{max}} \quad (16)$$

B. Detection of star topology

LeaderRank skewness $\hat{\nu}$ can be utilized to indicate adverse leader ranking by means of LeaderRank scores. In this section, we compare ν to the classic measure of network entropy (denoted as H in the following text). In order to allow direct comparison to $\hat{\nu}$ as well as to entropies computed from other graphs, H is required to be normalized analogously to $\hat{\nu}$. In this subsection, we give a brief overview on how normalization can be conducted.

Let A be the adjacency matrix of a network with N vertices, where each element $a_{ij} := 1$ if there exists a directed edge e_{ij} between adjacent vertices v_i and v_j. Each element of the principal diagonal a_{ii} is defined as $a_{ii} := \deg(v_i)$ and thus corresponds to the degree – the sum of the incoming and outgoing edges – of vertex v_i. The trace of A is defined as the sum of all elements of the principal diagonal: $tr(A) = \sum_{i=1}^{N} a_{ii}$. The formalism for graph entropy used by Passerini and Severini $H(\rho) = -tr(\rho \log_2 \rho)$ [25] is based on the von Neumann entropy and can be adapted as shown in

(17).

$$\begin{aligned} H(\rho) &= -tr(\rho \log_2 \rho) \\ &= -\sum_{i=1}^{N} \rho_i \log_2 \rho_i \\ &= -\sum_{i=1}^{N} \frac{a_{ii}}{tr(A)} \log_2 \frac{a_{ii}}{tr(A)} \\ &= -\sum_{i=1}^{N} \frac{\deg(v_i)}{\sum_{j=1}^{N} \deg(v_j)} \log_2 \frac{\deg(v_i)}{\sum_{j=1}^{N} \deg(v_j)}. \end{aligned} \tag{17}$$

This formalism, which is the entropy of the density matrix of a graph, describes the distribution of incoming and outgoing edges. In a randomly generated graph one expects $\deg(v_i) \approx \deg(v_j)$. In this case, the graph entropy H is close to the theoretical maximum entropy H_{max}. Therefore, the graph entropy only reaches its maximum if G is a regular graph where $\deg(v_i) = \deg(v_j) = D$. Because $\rho_i = D/DN = 1/N$ in a regular graph, one has H as shown in (18).

$$H = H_{max} = -\sum \rho_i \log_2 \rho_i = \log_2 N \tag{18}$$

In contrast, the minimum graph entropy H_{min} is observable in networks showing star topology. The trace $tr(A)$ of such a graph corresponds to $2N-2$ and the degree of its central vertex is $\deg(v_c) = N-1$. Consequently, the entropy of the central vertex H_c is calculated as shown in (19).

$$H_c = -\frac{N-1}{2N-2} \log_2 \frac{N-1}{2N-2} = -\frac{1}{2} \log_2 \frac{1}{2} = 0.5. \tag{19}$$

The degree of any other vertex is $\deg(v_i) = 1$. Hence, the entropy of a graph constituted as a star is calculated as follows:

$$\begin{aligned} H &= H_{min} \\ &= 0.5 + \sum_{V \setminus v_c} -\frac{1}{2N-2} \log_2 \frac{1}{2N-2} \\ &= 0.5 + \frac{1}{2} \log_2 (2N-2) \\ &= 1 + \frac{1}{2} \log_2 (N-1). \end{aligned} \tag{20}$$

The normalized network entropy can be finally computed according to (21):

$$\hat{H} = \frac{H - H_{min}}{H_{max} - H_{min}}, \hat{H} \in [0,1] \tag{21}$$

In order to illustrate expressiveness of $\hat{H}$ and $\hat{\nu}$ with respect to the underlying network topology, a straightforward experiment was carried out in which synthetic networks exhibiting star topologies were continuously mutated over time, resulting in almost regular graphs after numerous generations.

This simulated process consequently yields a continuous change of network topology for each graph. $\hat{H}$ and $\hat{\nu}$ were accordingly computed for every generation and tracked. The time series of both measures are shown in Figure 1. More precisely, simulations of topological change were conducted by starting with star graphs of fixed sizes ($N = 16, 32, 64, 128, 256$ and 512 vertices). In every generation, edges between every pair of vertices were randomly added and respectively removed.

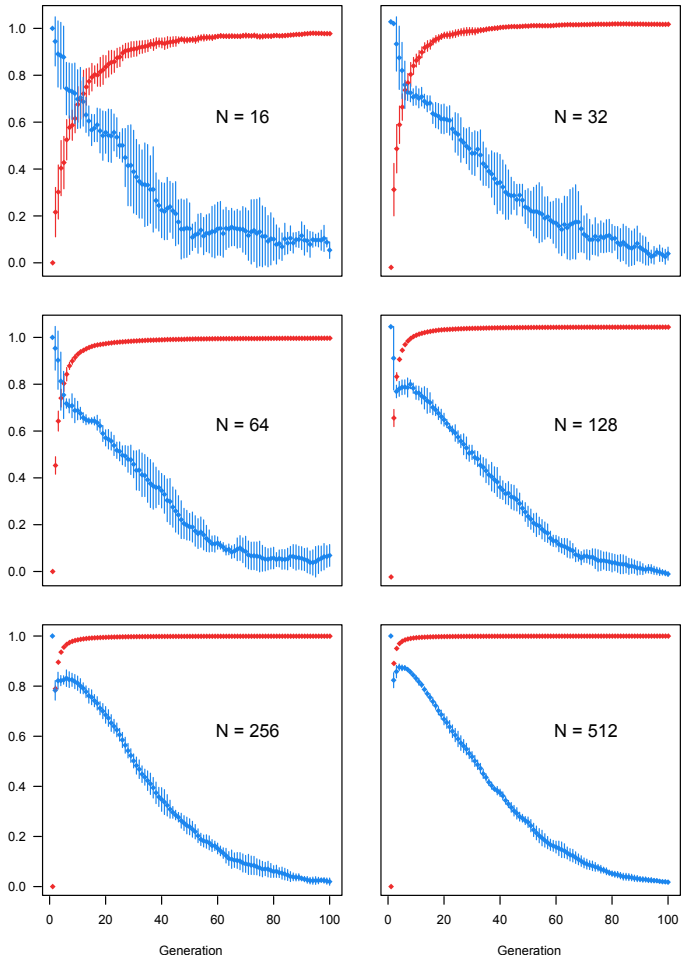


Figure 1. Simulation results of networks with various sizes N, whereas the red line represents $\hat{H}$, the blue line $\hat{\nu}$ and vertical bars indicate standard deviations.

For each graph size, six runs were conducted in an effort to estimate variance.

As shown in Figure 1, both measures converged after 100 generations. All entropy trajectories show fast convergence compared to $\hat{\nu}$ trajectories, with the convergence time decreasing with increasing N. Although $\hat{\nu}$ yield larger variances (especially for $N \leq 32$), its slower convergence and qualitatively similar trajectories for all graph sizes N illustrates greater sensitivity to topological changes. In that respect, matrix entropy loses significance with increasing graph size.

IV. Datasets

In this study, two different networks, namely Facebook and Twitter, of the German party "DIE LINKE" were analyzed, because both exhibit a star-like topology, yet to a different degree. As a comparison, a part of the Epinions social network, as an example for a nearly regular graph, was also included.

A. Facebook Dataset

Figure 2 depicts the network of the Facebook page "DIE LINKE" from January 2017 as a graph in which the size of each node corresponds to the out-degree (number of out-links). As can be seen, the network is dominated by the central node

of the page owner and, therefore, closely resembles a star-shaped topology.

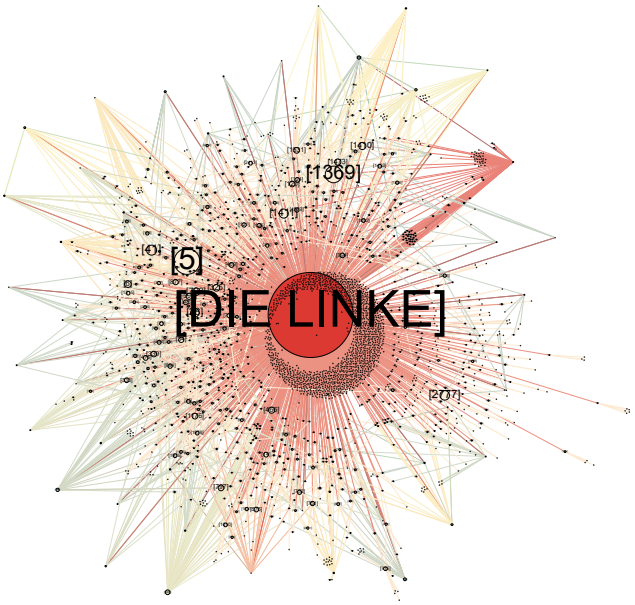


Figure 2. The network of the facebook page "DIE LINKE" of January 2017.

The central node often has the highest activity, meaning the most in- and out-links. The communication on the page was explored over a period of five months, from January 2017 up until May 2017, whereas all posts, comments and replies were taken into account as can be seen in Table I.

TABLE I. SUMMARY OF THE DATA INCLUDING NORMALIZED ENTROPY AND SKEWNESS OF THE CONSIDERED NETWORKS.

month	actors	posts	comments	replies	$\hat{H}$	$\hat{\nu}_{LR}$
January	2,878	26	2,955	3,471	0.19	0.98
February	2,146	33	2,196	2,062	0.24	0.98
March	3,196	40	3,501	3,245	0.17	0.97
April	2,432	26	2,558	3,295	0.22	0.98
May	4,765	31	4,130	5,674	0.10	0.98

Furthermore, it shows the normalized entropy and LeaderRank skewness of the "DIE LINKE" network, separately calculated for each month. It can be clearly seen that obtained $\hat{H}$ values fluctuate over time, whereas the LeaderRank skewness $\hat{\nu}_{LR}$ remains stable.

During the initial analysis of the dataset, it was observed that $12,031$ individuals were active throughout the five months. However, as shown in Figure 3, only 104 of these individuals were active in every single month. In general, it can be stated that users showed rather sparse and sporadic activity, with only a minority being recurrent users. Thus, yet again, this supports the assumption this network has a star-like topology. Additionally, this may indicate that the activity of users and, subsequently, the degree of opinion leadership, depends on the topics being discussed in a certain time period. However, in order to support this claim, further analyses need to be undertaken, which will be covered in a future study.

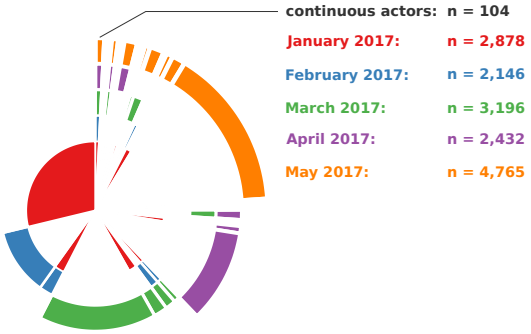


Figure 3. Sunburst chart of actor activity in the Facebook network consisting of one radial segment for each user, whereas a user's segment in a time layer is left out if said user was observed to be inactive in that time period.

B. Twitter Dataset

In a subsequent analysis the Twitter network of "DIE LINKE" was evaluated.

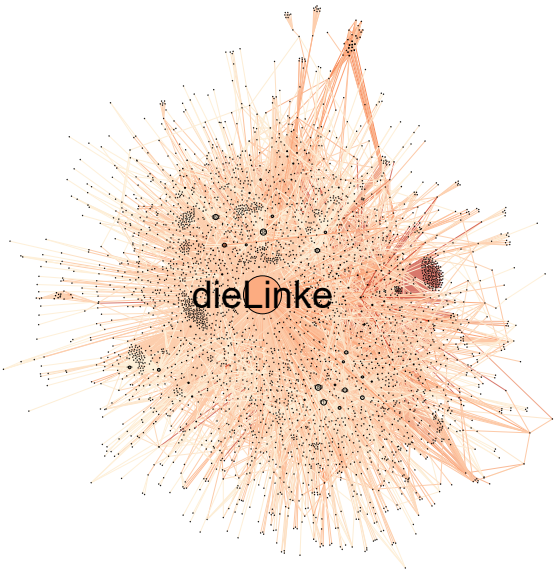


Figure 4. Twitter network of "DIE LINKE" of January 2018.

As can be seen in Figure 4 the star topology is less predominant for this network in comparison to the Facebook dataset. Consequently, a star topology is recognizable but at the same time some accounts besides "DIE LINKE" emerge.

The Twitter data set consists of tweets authored by "DIE LINKE", tweets addressing "DIE LINKE" and replies to the respective tweets. Aforementioned data was collected for the entire year of 2018 and on average twice as many actors where involved in the network compared to the Facebook data.

With an $\bar{\hat{\nu}} = 0.73$ the statistical analysis of the data shows that even though the star-like topology is not as distinctive as for the Facebook network it is still relatively strong as could already be seen in Figure 4. Furthermore, in comparison to the Facebook network the values for the skewness in the Twitter network show a greater fluctuation or to be precise cover a

TABLE II. SUMMARY OF THE TWITTER DATA INCLUDING NORMALIZED ENTROPY AND SKEWNESS.

month	actors	tweets	$\hat{H}$	$\hat{\nu}_{LR}$
January	5,966	10,695	0.39	0.74
February	6,194	11,466	0.40	0.79
March	7,677	14,820	0.44	0.86
April	7,179	12,711	0.38	0.84
May	7,529	14,349	0.36	0.77
June	8,864	21,407	0.14	0.86
July	6,612	13,951	0.22	0.67
August	6,834	13,033	0.24	0.79
September	8,072	16,631	0.33	0.79
October	6,943	13,974	0.26	0.87
November	5,757	10,249	0.32	0.76
December	5,642	9,119	0.38	0.75

greater range ($R^{FB}_{\hat{\nu}} = 0.1, R^{T}_{\hat{\nu}} = 0.2$). However, they are still more stable than the corresponding values for the entropy ($R_{\hat{H}} = 0.3$).

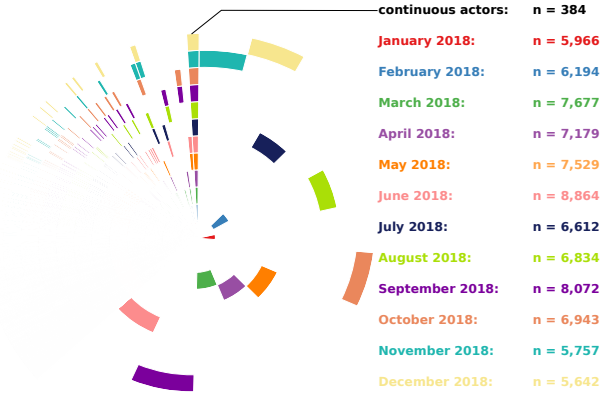


Figure 5. Sunburst chart of actor activity in the twitter network consisting of one radial segment for each user, whereas a user's segment in a time layer is left out if said user was observed to be inactive in that time period.

As can be seen in Figure 5, similar to the Facebook network, only a small amount of users is active throughout the entire year, yet rather their activity is concentrated on certain months.

C. Epinions Dataset

Figure 6 shows part of the *Epinions* social network [26] which, in contrast to the previous datasets, tends to be regular. Subsequently, there is no node, which dominates all others in terms of its degree. In this figure, due to the size of the network, it was necessary to arbitrarily limit the depiction by applying k-core ≥ 80 [27] showing only the most active nodes.

In comparison to the other networks, the *Epinions* social network [26] consisting of 75,879 actors shows a normalized network entropy $\hat{H} = 0.65$ and a normalized leader rank skewness $\hat{\nu}_{LR} = 0.07$, indicating a considerably less skewed LeaderRank score distribution.

The three discussed real world examples support the results of the simulation experiment discussed in Section III, whereas the normalized network entropy is less expressive in regards to an evaluation of the network topology than the LeaderRank skewness.

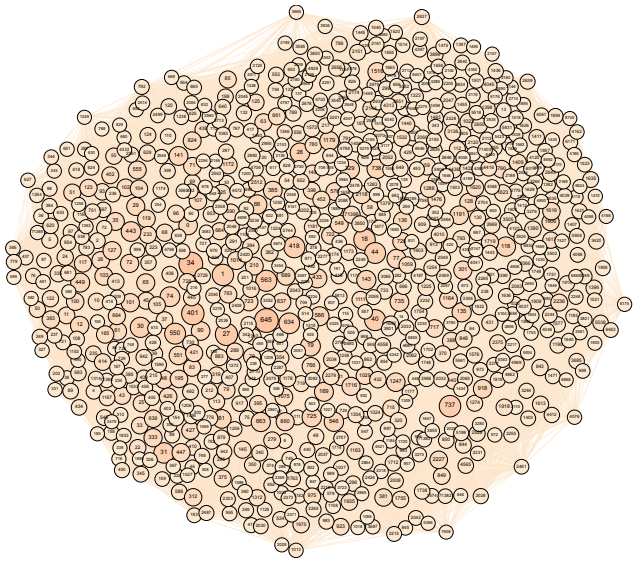


Figure 6. Part of the *Epinions* social network [26] (filtered by k-core ≥ 80).

V. COMPETENCE BASED RANKING APPROACHES

To address the issues discussed in Section III, we present a modification of the original LeaderRank referred to as CompetenceRank as well as some additional heuristics as improvements to incorporate specific features found in social networks.

A. CompetenceRank

In order to counteract the skewness of the LeaderRank in graphs with a star-like topology, the LeaderRank score of actors with a high degree of interaction, who at the same time only receive minimal attention by others, needs to be penalized. Similar to a citation network the relevance of a vertex does not only depend on the number of its interactions, yet it rather depends on a balanced ratio of own interactions and references by others. If this ratio is used as a weighting of the LeaderRank only those actors remain in the top ranks whose influence is based mainly on their competence.

Therefore, let V be the set of all vertices representing the actors of a social network and E be the set of all directed edges representing the relationship between vertices for example the communication or followers. The CompetenceRank CR_i of a particular actor $v_i \in V$ lowers the LeaderRank score S_i depending on the ratio of out-going and in-coming edges.

$$CR_i = \frac{S_i}{1 + \frac{e^{out}_{v_i}}{|E|} \sum_{v \in V} S_v} \tag{22}$$

The CompetenceRank as shown in (22) is subsequently calculated by dividing the original LeaderRank score S_i by a fraction of the cumulative sum of LeaderRank scores defined by the vertex's share of network activity, with $e^{out}_{v_i}$ being the number of its outgoing edges. By definition, the sum of LeaderRank scores of all vertices in the social network graph is equal to the number of actors N. When considering regular graphs, one observes LeaderRank distribution skewness $\hat{\nu} = 0$ as well as $e^{out}_{v_i} = e^{out}_{v_j} = D$ for any pair of randomly chosen vertices v_i and v_j. Thus, $|E| = ND$. From this, (22) can be

rewritten as

$$CR_i = \frac{S_i}{1 + \frac{D}{ND}N} = \frac{1}{2}S_i. \quad (23)$$

We finally define the CompetenceRank based on the assumption that $S_i = CR_i$ in regular graphs which is thus simply achieved by multiplying the expression in (23) by 2 as given in (24).

$$CR_i = \frac{2S_i}{1 + \frac{e_{v_i}^{out}}{|E|}N} \quad (24)$$

As shown in (25) one can calculate the discrepancy between the CompetenceRank and the LeaderRank in terms of the root-mean-square deviation $RMSD$ (note: vertical line denotes average sum). In turn that value can be seen as a further function of network regularity besides the measures discussed in Section IV.

$$RMSD = \sqrt{\sum_{i=1}^{N} [CR_i - S_i]^2} \quad (25)$$

On average one receives an $RMSD$ of 11.3 for the Facebook network, of 5.7 for the Twitter network and of 4.9 for the Epinions network.

B. Improved CompetenceRank

Especially social networks include many additional features that support the idea of a competence based ranking. In particular *likes* and *shares* play a special role in social media and reflect the acceptance of an expressed opinion and, as a consequence, should be considered when assessing the competence. Neither the LeaderRank nor the CompetenceRank as reported in [1] take these features into consideration or are even designed to include additional features. In the following paragraphs heuristics of the most important features of social network are designed and step by step combined in a weighted manner in order to reflect the relevance of different features regarding the competence in various types of social media platforms.

1) Pivoted post frequency normalization: As already discussed, if an actor posts messages with a high frequency without receiving much response from the network, their activity becomes less valuable and their LeaderRank score needs to be lowered. This means, when looking at it from another point of view, the fewer messages an actor posts, while at the same time receiving great response from the rest of the network, the more valuable this actor becomes. Consequently, their score needs to receive a higher rank. How much the rank needs to be lowered or raised has to depend on how much the individual's posting frequency deviates from the average posting frequency of all actors in the network. A similar behavior was described by Singhal et al. in 1995/1996 [28] with the pivoted length normalization for the text retrieval problem.

$$normalizer = 1 - b + b\frac{PF_i}{\sum_{i=1}^{N} PF_i} \quad (26)$$

Its original core idea is to reward or penalize a document based on the document length in relation to the average document length within a given collection of documents. For social networks this easily adapts to rewarding or penalizing actors when their total activity is either above or below the average activity in a given network. This leads to the equation as shown in (26), whereas the total activity is measured by the post frequency PF_i of the individual actor v_i and b controls how much an actor's activity is rewarded or punished. Depending on the network, the extent to which the activity is rewarded or penalized differs. In general, achieving a high degree in opinion leadership within a network requires individuals to understand and conform to its code of conduct. For example, when comparing a network of scientific publications and citations to a twitter network, then the former is defined by a rather low publication or post frequency but with a high quality whereas the latter favors a high activity but limits the depth and quality with a length limitation on each tweet. Moving from twitter to the scientific domain and vice versa inevitably requires an adaption to the new circumstances and only if this transition in behavior is achieved will one be able to maximize their influence in the respective area. In summary, the pivoted post frequency normalization rewards individuals that maintain a post frequency in line with or higher than average.

2) Sublinear post frequency transformation: Especially in networks that tend to have a star-like topology, few very active actors dominate the entire network. In the field of information retrieval a similar problem is addressed with a sublinear term frequency transformation, whereas one of the most popular approaches is Robertson's BM25 [29]. Here, the gain is lowered with an increasing term-frequency, while, at the same time, an upper bound of the term frequencies is defined. When adapted to the problem of highly active actors in social networks the impact of increasing posting frequencies can be lowered and with $k + 1$ an upper bound can be defined as shown in (27).

$$gain = \frac{(k+1)PF_i}{k + PF_i} \quad (27)$$

As previously discussed, the degree of opinion leadership partially relies on respecting the circumstances. While the pivoted post frequency normalization ensures that low activities are being penalized it also offers the chance of a disproportionate reward for users that are drastically more active than average. Therefore, the sublinear post frequency transformation diminishes returns that result from high activity and introduces an upper limit that prevents actors from extensively receiving a disproportionate gain. This concept allows users to benefit from being slightly more active than average while at the same time approaching the upper boundary requires a significant increase in activity.

3) Post frequency normalized LeaderRank: Using a combination of the pivoted post frequency normalization and the sublinear post frequency transformation as a weight for the LeaderRank score leads to a post frequency normalized LeaderRank nS_i as shown in (28).

$$nS_i = S_i \left[1 - b_1 + b_1 \frac{\frac{(k_1+1)PF_i}{k_1+PF_i}}{\sum_{i=1}^{N} PF_i}\right] \quad (28)$$

Using this equation the original LeaderRank is weighted by a fraction of an actor's activity in the entire activity of all network actors, whereas with k_1 the dominance of extreme activity over all other activities is minimized. Furthermore, with b_1 it is possible to control how much the degree of activity above or below the average is punished or rewarded,

respectively. The normalized LeaderRank as shown in (28) has a similar effect as the CompetenceRank in (24). However, with the parameters it is possible to adjust the normalized LeaderRank to the conditions of a specific network. For example, for a platform that focuses on posts with a high quality one may chose a low value for b_1, because of the low importance of the post frequency. Contrarily, for a platform like Twitter, which focuses more on activity, a higher a value can be chosen.

4) Incorporating likes and shares: likes are a key aspect of social media platforms as they show the acceptance of an actor by other actors and are, thus, an expression of competence. Therefore, they need to be taken into account when evaluating the impact of any given individual on such a platform and the post frequency normalized LeaderRank is combined with the average number of *likes* $\frac{LF_i}{PF_i}$ a user receives per post, whereas LF_i denotes the like frequency of a certain actor. Averages are used since a user could, for example, have an especially high number of 300 posts and only acquire one *like* per posted message. Contrary, a user posting three times could be receiving 100 *likes* per message. The total *like* count might be similar, yet the impact of the former appears to be marginal while the content of the latter seems to be well received and quite influential. In contrast to other activities in a network, such as creating posts, normally *likes* are connected to a post or message and not a certain actor. This means that everyone who can read the post can *like* it, even though they might not be part of the observed network and it is impossible to ensure that only those *likes* are considered that are from actors also active in this network. For example, in the "DIE LINKE" Twitter network, a tweet from a member of a right-winged party could appear in the network if it is directed to "DIE LINKE". This tweet might receive a lot of attention from other actors, active in the right-winged party network, yet not much attention from actors of the twitter network "DIE LINKE". Nonetheless, the tweeting actor would receive a high number of average *likes* for the "DIE LINKE" Twitter network. A similar effect could be achieved if *likes* are received through bots or are bought. Therefore, the normalized like score nLS_i of an actor v_i is calculated as the average number of *likes* this actor's posts receive weighted with the fraction of the actor's activity in the overall activity of the network as shown in (28).

$$nLS_i = \frac{LF_i nS_i}{PFv_i \sum_{i'=1}^{N} nS_{i'}} \tag{29}$$

Another important aspect of social networks is the number of posts by an actor that have been shared by other actors. In comparison to the number of *likes* a post receives, a highly shared post/tweet extends its reach significantly, consequently allowing the individual to influence more actors than they normally could. Similarly to (28) concepts like pivoted shares frequency normlization and sublinear shares frequency transformation are utilized together with their parameters k_2 and b_2 to maintain a controlled environment without too heavily benefiting extreme cases, resulting in a normalized share score nSS_i as shown in (30), whereas $\overline{SF}_i$ denotes the average share frequency an actor v_i receives.

$$nSS_i = 1 - b_2 + b_2 \frac{\frac{\left[1+k_2 \sum_{i'=1}^{N} \overline{SF}_{i'}\right]\overline{SF}_i}{k_2 \sum_{i'=1}^{N} \overline{SF}_{i'} + \overline{SF}_i}}{\sum_{i'=1}^{N} \overline{SF}_{i'}} \tag{30}$$

Finally, all components are combined resulting in the improved CompetenceRank CR_i as shown in (31) with α being the parameter that weights the normalized like score depending on the importance of *likes* in the observed network.

$$CR_i = [nS_i + \alpha nLS_i]\, nSS_i \tag{31}$$

VI. Results

Since the required additional data, i.e., *likes* and *shares*, were not available for the Facebook dataset, only the Twitter network of "DIE LINKE" for the year 2018 was analyzed with the new improved CompetenceRank and compared with the results of the LeaderRank. In the analysis, the parameter b was set to 0.7, k_1 was defined as the average tweet frequency in the entire network, k_2 as double the amount of the average tweet frequency and α was set to two assuming that liking is twice as important for competence as activity in the considered network.

An overview of the five highest opinion leader scores, indicating the discrepancy between the results for the LeaderRank and the improved CompetenceRank, is shown in Figure 7. As can be seen, the five accounts with the top scores are for the LeaderRank less diverse over the entire year as compared to the improved CompetenceRank. Lacking diversity in itself is not necessarily negative, however, when looking at the results for the LeaderRank it can be noticed that the accounts in Figure 7 include several political parties. Over the duration of 12 months, excluding the account of "DIE LINKE" (the owner of the network), with the LeaderRank it was possible to identify 19 accounts of possible opinion leaders, of which 9 belong to political parties (e.g. "afd", "cdu", "fdp", "linke_sh"). In comparison, a total of 23 accounts were identified using the improved CompetenceRank of which only 5 belonged to political parties.

It is not surprising that political parties appear in the top ranks, as they are a quintessential part of political discourse and thus it is their aim to shape the political opinion of the citizens. However, political parties reflect the consensual opinion of their members. Nevertheless, the ideas shaping the opinion of others and thus the political discourse as such often come from individuals. These opinions and ideas are not necessarily conform with the congruent opinion of the party. Still, they inspire the discussion and have the potential to influence the consensus. When only considering the activity of an account, as does the LeaderRank, such accounts, cannot compete with the accounts of political parties that are used to inform the public about the activity of the party and are thus highly active within a network. The improved CompetenceRank is able to raise the ranking of these accounts and to lower the ranking of those accounts that only receive a high rank because of their activity.

Deeper insight was provided by a thorough analysis of the monthly datasets. The LeaderRank and the improved CompetenceRank were calculated, providing us a total of two different ranked lists. Subsequently, to minimize the potential of performing well by chance on the first five accounts, the analysis of a list of five accounts per month was extended to the 20 highest ranking accounts per month. Ranked lists need to be evaluated in a way that reflects increased or decreased performance. Hence, the identified accounts were divided into six different categories: Individuals, Journalists, News, Political Parties, Politicians, Other and Unknown.

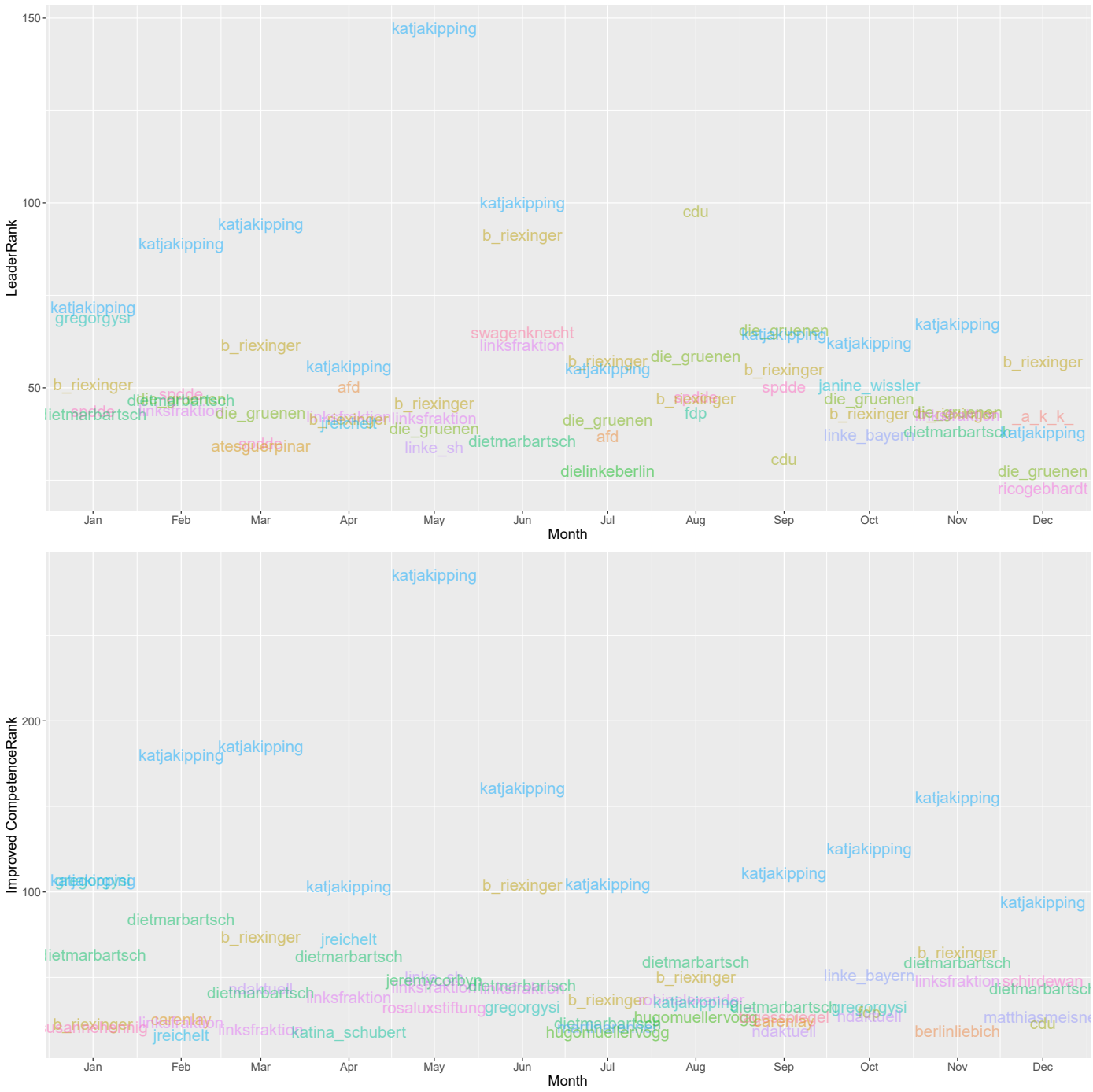


Figure 7. Comparison of the fivea accounts with the highest LeaderRank (upper graph) and improved CompetenceRank (lower graph) scores for the year 2018 for the Twitter Network of “DIE LINKE” .

Each of the 20 accounts per month received a label that was derived through manual evaluation of their Twitter profiles. Ordinary Twitter accounts, seemingly run by individuals without an obvious political office or a position in journalism were labeled as “Individual”. Following this procedure individuals with an obvious background in the field of the news industry were labeled as “Journalists”, whereas accounts tweeting on behalf of a news organization, accordingly do not represent the opinion of a single individual hence giving them the label “News”. In the same manner “Politician” refers to a single individual being either active in a political party or involved in a political office. Analogously to “News”, “Political Party” refers to an account tweeting on behalf of a political party. “Other” includes everyone not fitting into previously mentioned categories (e.g. companies, bands, NGOs etc.) and finally “Unknown” includes suspended and deleted accounts. The total results are displayed in Table III.

In the given network, it becomes evident that political

TABLE III. SUMMARY OF TYPE OCCURRENCES FOR MONTHLY DATA SUMMED UP ACCORDING TO THE USED MEASURE

Type	LR	Improved CR
Individual	10	25
Journalist	2	26
News	11	30
Politician	99	115
Political Party	104	36
Other	11	4
Unknown	3	4

parties are down-ranked according to their influence by the improved CompetenceRank, whereas individuals, journalists and news outlets receive higher ranks. This result confirms the assumption that the improved CompetenceRank counteracts the skewness of star-shaped topologies, as can be found for example in political networks, and further allows to distinguish the real initiators that trigger the intraparty pattern of opinions from the mass of other unimportant accounts in the network.

Furthermore, an account identified as an opinion leader should be associated with a small group or even a single person. This can be brought back to Katz' original thesis that large parts of society are not influenced by mass media or in our specific case by organizations and political parties but rather by trustworthy, influential opinion leaders. In turn, identifying 36 instead of 104 political parties is a considerable improvement, because it allows to identify more individuals, more journalists and more politicians. Moving away from pointing out the general importance of political parties and instead selecting specific individual accounts exerting their influence over a given social network is of tremendous value.

In this experiment the improved CompetenceRank outperforms the LeaderRank as it returned fewer political parties, fewer accounts of category "Unknown" and fewer suspended or deleted accounts. The analyzed Twitter network is less skewed than the Facebook network, as shown in Section IV. Therefore, it can be assumed that the improvements become even more distinctive when analyzing a highly skewed network.

VII. CONCLUSION AND FUTURE WORK

The analysis of social networks, and in particular identifying influential and opinion-influencing profiles, is of great interest in forensic research for a variety of reasons. In the present study, it was shown that the usual centrality-based approaches, and in particular the LeaderRank, produce erroneous results in star-like networks, such as Facebook pages of political parties. Furthermore, LeaderRank skewness was presented as an appropriate measure to quantify the degree of distortion of a network or in other words its proximity to a star-shaped topology.

Subsequently, CompetenceRank was introduced as a measure to overcome the shortcomings of the popular LeaderRank in star-like network topologies.

Additionally, an improvement of the CompetenceRank was provided incorporating fundamental interaction data such as "likes" and "shares". This methodology was tested on the Twitter network of "DIE LINKE". Identifying political parties as dominant and influential accounts on social media does not yield significant new insight into a political network since the importance of such accounts can be derived prior to any analysis as political discussions are frequently centered around political parties. However, pointing out influential individual politicians or individuals in general aligns more with the goal and image one has in mind when talking about an opinion leader. It was shown that the new measure outperforms the LeaderRank by identifying considerably more individual Twitter accounts and attributing less importance to accounts run by political organizations.

In following studies, it would be interesting to analyze the observed phenomena in more fine-grained time ranges. Additionally, it is necessary to take more and different network topologies into account. Furthermore, it was noticed that the texts in the Facebook data used were surprisingly well written. This provides an opportunity to conduct further textual analyses especially to answer the question whether there is a correlation between topics and opinion leaders and if so, how both develop over time.

REFERENCES

[1] M. Spranger, F. Heinke, H. Siewerts, J. Hampl, and D. Labudde, "Opinion Leaders in Star-Like Social Networks: A Simple Case?" in The Eighth International Conference on Advances in Information Mining and Management (IMMM). Barcelona, Spain: IARIA, July 2018, pp. 33–38.

[2] E. Katz, "The two-step flow of communication: An up-to-date report on an hypothesis," Public Opinion Quarterly, vol. 21, no. 1, Anniversary Issue Devoted to Twenty Years of Public Opinion Research, 1957, p. 61.

[3] M. Spranger, F. Heinke, S. Grunert, and D. Labudde, "Towards predictive policing: Knowledge-based monitoring of social networks," in The Fifth International Conference on Advances in Information Mining and Management (IMMM 2015), 2015, pp. 39 – 40.

[4] M. Spranger, H. Siewerts, J. Hampl, F. Heinke, and D. Labudde, "SoNA: A Knowledge-based Social Network Analysis Framework for Predictive Policing," International Journal On Advances in Intelligent Systems, vol. 10, no. 3 & 4, 2017, pp. 147 – 156.

[5] M. Spranger, S. Becker, F. Heinke, H. Siewerts, and D. Labudde, "The infiltration game: Artificial immune system for the exploitation of crime relevant information in social networks," in Proc. Seventh International Conference on Advances in Information Management and Mining (IMMM), IARIA. ThinkMind Library, 2017, pp. 24–27.

[6] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," Comput. Netw. ISDN Syst., vol. 30, no. 1-7, Apr. 1998, pp. 107–117.

[7] L. Lü, Y.-C. Zhang, C. H. Yeung, and T. Zhou, "Leaders in social networks, the delicious case," PloS one, vol. 6, no. 6, 2011, pp. 1–9.

[8] Q. Li, T. Zhou, L. Lü, and D. Chen, "Identifying influential spreaders by weighted leaderrank," Physica A: Statistical Mechanics and its Applications, vol. 404, no. Supplement C, 2014, pp. 47 – 55.

[9] E. M. Rogers, Diffusion of innovations. New York: The Free Press, 1962.

[10] B. Huang, G. Yu, and H. R. Karimi, "The finding and dynamic detection of opinion leaders in social network," Mathematical Problems in Engineering, vol. 2014, 2014, pp. 1–7.

[11] P. Parau, C. Lemnaru, M. Dinsoreanu, and R. Potolea, "Opinion leader detection." in Sentiment analysis in social networks, F. A. Pozzi, E. Fersini, E. Messina, and B. Liu, Eds., 2016, pp. 157–170.

[12] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," J. Mach. Learn. Res., vol. 3, Mar. 2003, pp. 993–1022.

[13] X. Song, Y. Chi, K. Hino, and B. Tseng, "Identifying opinion leaders in the blogosphere," in Proceedings of the sixteenth ACM conference on Conference on information and knowledge management - CIKM '07, M. J. Silva, A. O. Falcão, A. A. F. Laender, R. Baeza-Yates, D. L. McGuinness, B. Olstad, and Ø. H. Olsen, Eds. New York, New York, USA: ACM Press, 2007, pp. 971 – 974.

[14] A. Aleahmad, P. Karisani, M. Rahgozar, and F. Oroumchian, "Olfinder: Finding opinion leaders in online social networks," Journal of Information Science, vol. 42, 09 2015.

[15] Y. Chen, L. Hui, C. I. Wu, H. Liu, and S. Chen, "Opinion leaders discovery in dynamic social network," in 2017 10th International Conference on Ubi-media Computing and Workshops (Ubi-Media), 2017, pp. 1–6.

[16] L. Jain and R. Katarya, "Discover opinion leader in online social network using firefly algorithm," Expert Systems with Applications, vol. 122, 2019, pp. 1 – 15. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S095741741830811X

[17] B. Amor et al., "Community detection and role identification in directed networks: Understanding the twitter network of the care.data debate," CoRR, vol. abs/1508.03165, 2015.

[18] M. Richardson and P. Domingos, "Mining knowledge-sharing sites for viral marketing," in Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining, Za&iuml and O. R. ane, Eds. New York, NY: ACM, 2002, p. 61.

[19] W. Oueslati, S. Arrami, Z. Dhouioui, and M. Massaabi, "Opinion leaders' detection in dynamic social networks," Concurrency and Computation: Practice and Experience. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/cpe.5692

[20] C. Egger, "Identifying key opinion leaders in social networks: An approach to use instagram data to rate and identify key opinion leader for a specific business field," Master Thesis, TH Köln - University of Applied Sciences, Köln, 2016.

[21] M. Z. Shafiq, M. U. Ilyas, A. X. Liu, and H. Radha, "Identifying leaders and followers in online social networks," IEEE Journal on Selected Areas in Communications, vol. 31, no. 9, 2013, pp. 618–628.

[22] Z. H. Zhang, G. P. Jiang, Y. R. Song, L. L. Xia, and Q. Chen, "An improved weighted leaderrank algorithm for identifying influential spreaders in complex networks," in 2017 IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), vol. 1, July 2017, pp. 748–751.

[23] R. Nadeau, E. Cloutier, and J.-H. Guay, "New evidence about the existence of a bandwagon effect in the opinion formation process," International Political Science Review, vol. 14, no. 2, 1993, pp. 203–213.

[24] R. M. Raafat, N. Chater, and C. Frith, "Herding in humans," Trends in Cognitive Sciences, vol. 13, no. 10, 2009, pp. 420 – 428.

[25] F. Passerini and S. Severini, "Quantifying complexity in networks: The von neumann entropy," Int. J. Agent Technol. Syst., vol. 1, no. 4, Oct. 2009, pp. 58–67.

[26] M. Richardson, R. Agrawal, and P. Domingos, "Trust management for the semantic web," in The Semantic Web - ISWC 2003, D. Fensel, K. Sycara, and J. Mylopoulos, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 351–368.

[27] G. D. Bader and C. W. V. Hogue, "An automated method for finding molecular complexes in large protein interaction networks," BMC bioinformatics, vol. 4, January 2003, p. 2.

[28] A. Singhal, C. Buckley, and M. Mitra, "Pivoted document length normalization," in Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ser. SIGIR '96. New York, NY, USA: Association for Computing Machinery, 1996, pp. 21–29.

[29] S. E. Robertson and S. Walker, "Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval," in Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ser. SIGIR '94. Berlin, Heidelberg: Springer-Verlag, 1994, pp. 232–241.

Extended Definition of the Proposed Open Standard for *IoT* Device Identific*A*tion and Reco*G*nition (*IoTAG*)

Lukas Hinterberger*
and Bernhard Weber†
Dept. Electrical Engineering and Information Technology
Ostbayerische Technische Hochschule
Regensburg, Germany
email:
lukas.hinterberger@st.oth-regensburg.de*
bernhard1.weber@st.oth-regensburg.de†

Sebastian Fischer
Secure Systems Engineering
Fraunhofer AISEC
Berlin, Germany
email:
sebastian.fischer@aisec.fraunhofer.de

Katrin Neubauer‡
and Rudolf Hackenberg§
Dept. Computer Science and Mathematics
Ostbayerische Technische Hochschule
Regensburg, Germany
email:
katrin1.neubauer@oth-regensburg.de‡
rudolf.hackenberg@oth-regensburg.de§

***Abstract*—Internet of Things (IoT) devices are critical to operate and maintain, because of their number and high connectivity. A lot of security issues concern IoT devices and the networks they are integrated. To help getting an overview of an IoT network, the devices and the security, we propose a scoring system to get a good impression of IT security. This system generates single scores for each device, using features like encryption, update behavior, etc. Furthermore, a summarized score for the whole network is calculated, to show the status of the network security in an easy way for the administrator. To enable the scoring system, a precise list of the existing devices and their operating status is necessary. To achieve this, we present an open standard for the *IoT* Device Identific*A*tion and Reco*G*nition (short *IoTAG*), which requires that devices report, e.g., their name, an unique ID, the firmware version and the supported encryption. The proposed standard is described in detail and an implementation guideline is given in this paper. Additionally, information about how to realize the serialization, the integrity and the communication with IoTAG.**

***Keywords*—Internet of Things; device identification; open standard; IoTAG; security rating.**

I. Introduction

This paper extends the already published paper "IoTAG: An Open Standard for IoT Device IdentificAtion and RecoGnition" [1] with more detailed information and definitions of the proposed open standard *IoT* Device Identific*A*tion and Reco*G*nition (IoTAG) and a guideline for the implementation.

Internet of Things (IoT) continues to be an innovation topic and trend in the industrial sector and smart homes. The development of new IoT devices, systems and services are progressing extremely fast. This raises the problem that the security risks of IoT devices, networks and services are underestimated or not even taken into account at all. It is precisely the reason that leads to insecure devices. An example of this would be the missing encryption or authentication. Some of these risks and vulnerabilities lead to attacks such as destroy the device. Serious attacks can lead to hijacking of complete company networks. In general, a large number of IoT devices are critical to operate [2] [3].

There are some solutions to these security problems. For example, with device detection, it should be possible for the user to detect devices in the IoT network and also check the software status. At present, there are no existing frameworks, software or systems for automated device scanning. With individual steps, it is possible to obtain individual units or parts of the required information. For example, addressable network ports can be found with the network scanners Nmap [4] or Fing [5]. The problem with Nmap or Fing is that the result of the scan will not be analysed or evaluated. Only a technical user or expert can perform and understand this technical analysis. A non-technical user needs a simple scoring system for IoT devices and networks.

The basic idea of our research project can be summarised as follows. The IoT devices of a network are identified and get a security rating during an initial scan. The rating is based on the provided metadata, information collected by the scanner itself and a database of known vulnerabilities, which are collected from multiple publicly available sources. An overall network rank results from the inheritance of the individual ranks. As part of the visual presentation for the end user (non-technical user), the rating should be shown as well as a list of all known vulnerabilities and general risks of the IoT setup.

The aim of the IoTAG project is to propose an open standard for IoT devices. This standard is intended to provide the required metadata for the risk and security rating and to verify the authenticity of the received information.

This paper begins with our hardware setup to test the idea of a network security score. Next, we started with the device scanning process and found out that it is not possible to get all the requirement information for our security evaluation and in some cases not even the device name or type at all.

We continue with the security criteria needed to create a device rating and then created the actual rating from this. As

a result, the entire network can be evaluated based on the individual scores.

As already discussed, the detection of the devices and their further details is not possible with available tools, we present a proposal for a standard which makes this possible. Since, the standard is still under development, a newer, more detailed version than in our last paper [1] is presented here. Furthermore, we have started with a sample implementation and give guidelines regarding the development.

The paper is structured as follows. Section II describes the related work. Section III introduces our hardware setup and device scanning, while Section IV defines the security criteria. Section V shows the device rating and Section VI the results. The standard IoTAG is presented in Section VII, followed by the conclusion in Section VIII.

II. Related Work

A popular solution for the identification of devices is the utilisation of so-called device fingerprints. Those can be used for basic categorisation and classification as secure or insecure. Miettinen et al. [6] show this procedure with device fingerprints for categorisation and classification. The development of an anomaly recognition system for smart home networks is taking place on the basis of a research project [7]. The subject device identification with device fingerprints and similar approaches covering by several publications [8]–[11]. The current working approach in the area of IoT device detection is shown. Currently, it is not possible to identify detailed information such as the current firmware version or a device ID for further recognition.

Khaled et al. [12] and Kaebisch et al. [13] proposing a machine readable description for IoT devices. These descriptions are not intended for risk and security device ratings. They are intended only for the functionality of a device and cover information like the turn off command. The goal of IoTAG is to get the security characteristics of an IoT device and no further information of the functions.

The Thing Description (TD) [14] by the World Wide Web Consortium (W3C) provides metadata of a device, e.g., stored setting or sensor data. An optional "Security" information on the authorization procedures is also available. But this is only a small part of all information needed for a security evaluation.

IoT Sentinel is a tool which detects and evaluates devices by creating a fingerprint and comparing it to a database of known devices. It is also able to isolate devices which are classified as insecure and filter their network traffic. The tool was developed by a team of researchers from the Technical University of Darmstadt, the Aalto University and the University of Helsinki [6]. In contrast to the commercial solutions discussed later in this Section, an example implementation is available under the MIT license, which allows for code reviews and further development [15].

Another approach is the security and privacy assessment for IoT devices with different security ratings. To calculate the device rating, this approach [16] uses the information protocols, open ports and encryption. This approach is very similar but it is not very flexible and user-friendly. The reason for this is the missing weighting of each criteria and the missing overall score of the network. Park et al. [17] and Ali et al. [18] show a list of security requirements for IoT services, which can be used as a basis for a risk assessment. These security requirements can be used to evaluate the weighting. A further approach to generate a metric value for the security of an IoT device is to use vulnerabilities and known exploits [19].

There are also multiple commercially available IoT security evaluation tools. One is Norton Core Router, which is developed by the anti-malware vendor Symantec Corporation [20]. Another one is Avira SafeThings, which is developed by Avira [21].

The scoring system, Norton Core Security Score, is deficit based, meaning it starts at 500 and each problem found reduces the score until it reaches the lowest score of 50. For example, if the firmware version of the router is outdated the score gets deducted by 10%. Not installing the client software "Norton Security" on a compatible device lowers the score. Most of the examples in the manual are not IoT related which indicates that the device detection is not detailed enough to provide the scoring algorithm with the needed information. One of the examples, which also applies to an IoT device, is ignoring an vulnerability or intrusion alert [22]. The vulnerability detection could be based on scanning for open ports and detecting the version of the software listening on them. This information could then be used to search for known vulnerabilities in that specific software, e.g., a web server. The exact way could not be examined, as the router was discontinued on January 31, 2019 and, according to the manufacturer, will only return as a software based solution in the future [23].

Avira lists a per-device security score as a feature. This score seems to be completely intransparent as it is neither mentioned nor described in the manual or any other resource about the device. Knowing how Avira classifies the individual devices, SafeThing should have enough information to give a helpful score, but as it is not described anywhere and as the device is currently unavailable for purchase, the scoring part could not be validated.

III. Hardware Setup and Device Scanning

The test environment consists of ten devices, as stated in Table I, which were selected to reflect a variety of typical IoT devices found in a home environment. A first basic network scan with nmap [4] resulted in a list of found devices and their hostnames. While some of the devices use meaningful hostnames, the list also contains a lot of generic names like "ESP" and empty rows. To gain more information, an extended scan, which includes a scan for open network ports, can be done as shown in Table II. This scan results in a list of found open ports and how the open port was found. Additionally, nmap lists the service which is registered for the found port at

the IANA (Internet Assigned Numbers Authority) [24]. This provides a first look at which services are used by the devices and how they communicate. For example, port 80 is specified to be used for http servers, which utilise unencrypted data transmission.

TABLE I. HARDWARE OVERVIEW

device	hostname
Amazon Echo 2	amazon-183e3c119
Apple iPhone 5	Kluges-iPhone
Floureon M32B	
Google Home mini	Google-Home-Mini
Grandstream GXP1610	
Raspberry Pi 3 Model B	raspberrypi
Sonoff Wi-Fi Smart Switch	ESP_6A768B
Wi-Fi Smart Bulb	ESP_4C3210
Wi-Fi Smart Plug	ESP_3D1EB6
Wi-Fi Touch Switch	ESP_469ACF

TABLE II. OVERVIEW OF OPEN AND RESTRICTED PORTS

Raspberry Pi 3 Model B				
port		**state**	**service**	**reason**
22	TCP	open	ssh	syn-ack
53	TCP	open	domain	syn-ack
Sonoff Wi-Fi Smart Switch				
port		**state**	**service**	**reason**
		restricted		
Wi-Fi Touch Switch				
port		**state**	**service**	**reason**
8081	TCP	open	blackice-icecap	syn-ack
Wi-Fi Smart Plug				
port		**state**	**service**	**reason**
10000	TCP	open	snet-sensor-mgmt	syn-ack
Grandstream GXP1610				
port		**state**	**service**	**reason**
22	TCP	open	ssh	syn-ack
80	TCP	open	http	syn-ack

After all, the information provided by these scans is still not enough to know the exact device model used in the network. For example, the running services on a device could vary based on the configuration of a device. The same applies to hostnames: there are no rules or limitations what devices can use as their hostname. Many devices, for example the iPhone, even allow the user to change it to a custom one.

IV. Security Criteria

For an automated security evaluation of an IoT network, a general applicable evaluation scheme is needed. The scheme has to be modular to allow for different devices being evaluated based on the used technologies. Every module is limited to a specific part of the device and the regarding security risks. The individual results can then be weighted against each other to obtain an overall evaluation of a device.

The scheme described below serves as a first approach for the evaluation of individual devices. It shall serve as a basis for the definition of the desired scan results and device properties and illustrate their later use.

TABLE III. SECURITY CRITERIA

audit criteria			**score**
radio technology			
WPA/WEP or no encryption			0
WPA2/WPA3			2
Bluetooth version			0-2
ZigBee version			0-2
manufacturer			
unknown manufacturer			0
usual patch time			0-2
experience			0-2
known unpatched devices			0-2
bug bounty program			0/2
services			
service	**default port**	**comment**	
HTTP	80	unencrypted login details	0
MQTT	1883	unencrypted control data	0
UPnP	49152/1900	firewall manipulation	0
rtsp	554	unencrypted video data	0
SIP	5060	unencrypted	0
service	**default port**	**comment**	
HTTPS	443	encrypted	2
MQTTS	8883	encrypted	2
SCP	10001	encrypted	2
SIPS	5061	encrypted	2
SSH	22	encrypted	2
LAN and WAN communication			
service	**default port**	**comment**	
HTTP	80	unencrypted login details	0
MQTT	1883	unencrypted control data	0
UPnP	49152/1900	firewall manipulation	0
rtsp	554	unencrypted video data	0
SIP	5060	unencrypted	0
service	**default port**	**comment**	
HTTPS	443	encrypted	2
MQTTS	8883	encrypted	2
SCP	10001	encrypted	2
SIPS	5061	encrypted	2
SSH	22	encrypted	2
other			
vulnerable to replay attacks			0
create own Wi-Fi			0
data retrieval without authentication			0
vulnerable to jamming			0-2
vulnerable to Denial of Service (DoS)			0-2
insecure configuration			0
continuous device number			0-2
known vulnerabilities			0
support lifetime			0-2
insecure / default password			0/2
firmware version			0-2
technical guidelines			0-2
certification			0-2

The aforementioned scheme utilises a three-value score system reaching from zero to two. If a module detects a critical security violation it results in a score of zero. A potential, but non-critical, violation would result in a score of one. If no problems are found, the score would be two. Similarly to the overall score calculation, each module runs several individual evaluations and weights them against each other to calculate the resulting score. A list of the modules can be found in Table III and are described in the following Subsections.

A. Physical connection

Although the software properties of a device play the main role for security risks, the physical connection to a

network could also be a potential attacking point. Therefore, a distinction is made between wired and wireless connections. If the connection is wireless, the used encryption technology is taken into account. A wireless connection results in a score of one, weighted against the score of the used encryption. A wired connection on the other hand is scored two, as physical access would be needed to interfere the connection.

Wireless connections can be unencrypted or use a variety of different encryption technologies. Obviously, the use of unencrypted or open Wi-Fi (wireless local area network) is considered dangerous and scored with zero points. The older Wi-Fi encryption standards, namely WPA (Wi-Fi Protected Access) and WEP (Wired Equivalent Privacy), are also scored zero points, as they use "RC4" (Rivest Cipher 4) for the encryption which is considered broken [25]. The use of the newer WPA2 and WPA3 standards results in the highest score of two.

B. Services

This module looks at the services, which are reachable from the network for the communication with a device. The rating of the security level is done for each listed service separately, but in this case the lowest individual score is used and not a weighted average. The evaluation is based on the protocol and encryption used. This is done using black and white lists. Services on the black lists either use obsolete protocols, which are considered vulnerable, vulnerable encryption or are completely unencrypted. A blacklisted service results in zero points, a whitelisted one in two points and if the service is not listed it is scored one point.

C. Communication

Besides the communication from the network, devices are also able to communicate by themselves to other devices. For example, many IoT devices connect to servers in the cloud or a local gateway. This communication is also evaluated based on the encryption used. As this type of connection cannot be detected by a scan of the network, the actual traffic needs to be analysed. This analysis also utilises a predefined list of protocols for the evaluation. Furthermore, the communication is split into LAN (local area network) and WAN (wide area network) in order to take the different security requirements in account. For example, data sent over the WAN leaves the relatively protected home environment and could therefore be seen by third parties. As both categories look quite similar, they are displayed as one in Table III.

In addition to the encryption, it is also possible to check the number of external resources a device communicates with and where they are located. An additional point that could be evaluated in the course of this analysis is whether a device requires a continuous connection to a cloud service. If no such service is used, two points could be are awarded. Otherwise, one point could be deducted.

D. Default passwords

When talking about passwords, a major security concern is the use of default credentials, which apply to all devices of the same type and manufacturer. If an attacker knows the credentials for a device, most of the other security measures are useless. Therefore, this module checks if a login with known credentials is possible. If it is the case, the score is set to zero. If the login was unsuccessful, the score is two, but is deducted by one if the username cannot be changed by the user.

E. Firmware version

Outdated firmware or unmaintained firmware increases the possibility of security vulnerabilities. Most known vulnerabilities are collected and provided to others in form of several vulnerability databases. CVE (Common Vulnerabilities and Exposures), for example, is one of the popular lists of known vulnerabilities, which is maintained by the MITRE Corporation and contains nearly 140000 entries [26]. As this information is also available to potential attackers, it allows for systematic attacks against outdated or unfixed firmware versions. Therefore, the module needs to be able to detect the installed version and check, if new versions are available. Additionally, it has to search vulnerability databases for known issues with the installed firmware version. If known vulnerabilities are found and no update is available, the lowest score of zero is given. If an update is available, the score is one and the user needs to be notified of this problem. If nothing of the mentioned applies, the device is up to date and is awarded with the highest score.

V. Device Rating

In this section, we describe the proceeding to receive the information for all the security criteria and how they are rated in detail.

A. Physical connection

In our test environment, a Raspberry Pi serves as a router through which all devices are connected to the network. For wireless and wired connections, different address spaces were used, which means that the physical connection of the individual devices can be determined via these address spaces.

The encryption technology of the wireless network can be taken from the router configuration. Since the software used for the access point is "hostapd", the configuration can be done in the "/etc/hostapd" file. The entry "wpa=2" indicates the exclusive use of the WPA2 standard, which in turn results in a score of two points for each device. If an insecure technology is used, this also affects the rating of each device, as the entire network is weakened. In this case, all devices in this category must be scored zero.

B. Services

The running services are recorded by scanning the individual network components. “Nmap” is used for this scanning process [27], which provides the results shown in Table IV.

TABLE IV. PORT SCAN

port	protocol
22	ssh
80	http
5060	sip

Based on these results and the categorization lists already mentioned, the device can be classified. The example in Table IV results in a score of 0.66 points, since http and sip are scored zero and ssh two points.

C. Communication

The communication of the devices with external resources is evaluated by recording and analyzing the network traffic. The MAC address of the local resource, source and destination port and the communication protocol are extracted from the communication packets by using the “tshark” software [28]. Incoming and outgoing traffic are analyzed independently of each other and similarly to the evaluation of the services, they are evaluated on the basis of predefined protocol lists. With the scan results shown in Table V, the device would be scored zero points.

TABLE V. COMMUNICATION SCAN

source device	destination port	protocol
00:11:22:33:44:55	5060	sip

The necessity of a cloud connection could not be checked automatically at this point, as it is not possible to determine whether a device is still fully functional after a possible interruption of such a connection.

D. Default passwords

Checking a device for the use of insecure login credentials is done using the software “THC-Hydra” [29], which performs a dictionary attack against the corresponding device. Both, the user name and the password are checked against known and frequently used terms. The information about the type of service for which a login check should be performed is taken from the previous service scan.

The use of non-standard logon procedures may cause a problem with this type of password check. For each specific procedure, a separate check algorithm would have to be developed, which might have to be adapted again after an update of the device firmware. An example of a vendor-specific login procedure is the challenge-response mechanism that AVM uses for the web interface of its Fritz!Box routers [30].

E. Firmware

In the absence of a standardized procedure for identifying the device firmware, it was not possible to check it in an automated procedure. The use of Nmap [4] allows assumptions about the operating system and other software components used on a device. However, due to rough inaccuracies, these are not sufficient for a valid risk evaluation. In addition, Nmap is only able to identify an operating system if it has already been fingerprinted in the past [31].

While it would be possible to create a Nmap fingerprint for each network device, this method can be considered irrelevant in practice because of the need to know the software running on each device. There is also no guarantee that the identifiers will not change after a software update, which would require the fingerprint to be recreated. These concerns can be transferred to procedures developed independently of Nmap.

F. Overall rating

After all the categories have been evaluated, an overall rating for a device can be calculated by averaging the ratings. This rating describes the vulnerability of a device based on Table VI. A ports score of 0.66 points, a communication score of 0.00 points, and a password score of 0.00 points results in an overall rating of 0.22 points, indicating that the device is highly vulnerable.

The security of the devices is indicated with an average evaluation instead of the lowest individual evaluation, since an overall impression of the device security should be given.

TABLE VI. VULNERABILITY CATEGORIES

score	category
0.00 to 0.80	high vulnerability
0.81 to 1.80	moderate vulnerability
1.81 to 2.00	small vulnerability

VI. RESULTS

For the verification and validation of the presented evaluation system, the following devices were tested as examples: Amazon Echo 2 (1), Apple iPhone 5 (2), Floureon M32B (3), Google Home mini (4), Renkforce RenkCast (5), Sonoff Wi-Fi Smart Switch (6), Wi-Fi Smart Bulb (7), Wi-Fi Smart Plug (8) and Wi-Fi Touch Switch (9). The results of the automatic evaluation related to these example devices can be found in Table VII. The devices were then checked manually and the resulting evaluation was compared with the automatically determined values. The values for “cloud only” and “default password”, shown in Table VII, where added manually. All the values are calculated as described in Section V.

In a later step, weightings can be assigned to the previously mentioned categories in order to make clear their different influences on device security.

The overall network rating is determined by the value of the least secure device.

TABLE VII. EXAMPLE RESULTS

parameter	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**
Wi-Fi encryption	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
services	1.00	2.00	0.33	1.00	2.00	2.00	1.00	2.00	2.00
LAN communication	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
WAN communication	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00	2.00
wired connection	1.00	1.00	2.00	1.00	1.00	1.00	1.00	1.00	1.00
cloud only	1.00	1.00	2.00	1.00	2.00	1.00	1.00	1.00	1.00
default password	2.00	2.00	1.00	2.00	2.00	2.00	2.00	2.00	2.00
overall score	**1.57**	**1.71**	**1.62**	**1.57**	**1.86**	**1.71**	**1.57**	**1.71**	**1.71**

While the calculation of the device ratings can be fully automated, it is not possible, as explained in Section III, to collect the necessary information automatically. For this reason, in the following chapters we present a new approach to retrieve device-specific information from the devices themselves.

VII. IoTAG Definition

The *IoT* Device Identific*A*tion and Reco*G*nition, short *IoTAG*, should be implemented into IoT devices by the manufacturers. It provides static and dynamic information about the device and its current state.

The focus of the IoTAG definition is on the standardized provision of critical device data, the integrity maintenance of the data sets to be transmitted and the relevance of the information for an individual classification of each device with regard to the implementation of security specifications and recommendations [32].

The core component of the IoTAG definition is the dataset specification. In the first instance, this is a list of attributes whose content is described. In the course of the technical specifications for the serialization of data for transmission over the network, data types and formatting specifications are assigned to these attributes.

In addition to a unique identifier for identifying individual devices, the dataset contains general product information such as a serial number, the device type according to fixed type definitions, a device category related to the main use scenario of the device, a product name and the manufacturer. Information about the installed hardware, such as the presence of a secure element or the use of a secure boot procedure, is also taken into account.

With regard to the connectivity of a device, information on the availability of LAN, WLAN or Bluetooth connections and their version is provided.

In addition to these static values, IoTAG also includes dynamic information about the update behavior of the device (automatic updates, end of support, etc.) as well as the device firmware (e.g., the current version). Furthermore, an overview of communication services such as SSH or HTTP servers, the associated software and the cryptographic algorithms that are used is provided.

In order to be able to associate the origin of the transmitted data with a device, a signature procedure will be presented, which is intended to ensure the integrity of this dataset. The signature is applied to the data serialized for communication as described later.

A. Dataset

The IoTAG Dataset consists of thirteen information about the IoT device:

1) Manufacturer
2) Name
3) Serial number
4) Type
5) ID
6) Category
7) Secure boot
8) Firmware
9) Client software
10) Updates
11) Cryptography
12) Connectivity
13) Services

1) Manufacturer: The manufacturer information is important to identify the device correctly and in case of a security issue, to contact the right company. The value of this information is a string that contains the company name as it is officially registered. This allows a clear assignment of the company, which is responsible for the device.

2) Name: The name is also a string, which contains the name of the device. It should be named as it is listed by the manufacturer with all the additional revision numbers like “Test Cam rev 3A”, to ensure an exactly identification in the case of security issues. Sometimes, not the complete batch of products is affected, because there could be a software update in later devices. This difference should be identifiable.

3) Serial Number: The next item, the serial number should be assigned by the manufacturer as a unique identification. It can be necessary for a network administrator to know all the serial numbers of his devices, if some of them are broken and need support or, if a security issue concerns some devices with a specific production date (which the manufacturer can identify by the serial number).

4) Type: To determine the potential damage of an attack, the device type is necessary. For example, a smart speaker cannot harm people directly. But if an attacker deactivates the smoke detector, it is a safety issue. The different types can help to estimate the damage and therefore, to separate the devices. We give some first suggestions for the type, but this list needs to be extended for all the different kind of IoT devices.

Suggestions for device types:

- Alarm system
- Camera
- Smart lock
- Smart speaker
- Smart TV

- Smoke detector
- Production machine
- Temperature sensor
- Security camera
- Emergency switch
- ...

5) ID: Besides the serial number, which is on required to be unique for one product, the identifier (ID) should be unique for every device worldwide. To achieve this requirement, the ID is created by concatenating the manufacturer name, the product name and the serial number. This string is hashed, using the SHA-256 algorithm [33] and encoded as base16 string [34], to ensure the right format. As for the use in IoT devices, the faster algorithm SHA2, prior to SHA3, is used [35] [36] [37]. The composition of the ID is shown below:

```
ID = BASE16 ( SHA-256 ( MANUFACTURER & PRODUCT NAME &
     SERIAL NUMBER ) )
```

6) Category: Similar to the type, the product category should help to determine the risk of an attack. But the category is not as accurate as the product type, because it should be used to categorise the different kind of products. Additionally, this can be used to separate the networks for the different device categories. Some examples are given in the following list:

- Assisted living
- Entertainment
- Household
- Industry
- Infrastructure
- Lighting
- Personal assistance
- Security
- ...

7) Secure Boot: The boolean value (true or false) for the secure boot indicates, if the device has a secure boot mechanism and therefore can ensure the integrity of its firmware at system startup.

8) Firmware: The firmware version is needed to check if there are new updates available. Additionally, an internet address must be given to download the newest version of a devices firmware. This is important, if the automatic update process is not working. Technically, the firmware is not one value, but two separate strings: the firmware version and a Uniform Resource Locator (URL) [38] to get the firmware. The version should consist of lexicographically ascending terms (higher number or character).

9) ClientSoftware: The client software is structured exactly like the firmware: version number and download URL. If a device does not use software for third-party devices, an empty string is returned.

10) Update: The update consists of multiple values. First, if the device updates itself automatically. This includes the whole process: check for new version, download and installation. It is an boolean value and named "Automatic updates".

The next value indicates, if the automatic update process is technically possible. If a connection to the update server can be established and the check and installation of updates is possible, it is set to "true". This value is also an boolean and named "Automatic updates possible".

The third value contains the date of the last update ("Last update on") and the fourth value the date of the end of support ("end of life") according to ISO 8601 [39].

11) Cryptography: To be able to make statements about the cryptographic capabilities of a device, it is necessary to know the algorithms used by the device. In addition, it must also be possible to make a statement as to whether these are implemented in hardware or software. It should also be specified whether secret keys are managed exclusively in secure hardware or in the main memory of the device.

The private key required for the signature of IoTAG as described in subsection C, is treated as a separate variable, as it is essential for the reliability of IoTAG.

Under the generic term cryptography, two identical structures are classified. Each of these subsections contains an attribute "IoTAG key", which is represented as a boolean value. If the key used for signature is managed exclusively in a secure hardware environment, the value "true" is assigned in the hardware structure and the value "false" in the software structure. If the key is accessible via software, the values are reversed.

Whether secret cryptographic keys are stored in any of the above-mentioned areas, is indicated by the boolean variable "key store". This variable can have the value "true" in both structures. The variable "algorithms" gives an overview of the cryptographic algorithms used in a device. This is a collection of character strings, which in turn represent a cryptographic algorithm according to its standardized name (example: "ecdsa-sha2-nistp256", defined in RFC 5656 [40]).

12) Connectivity: The physical possibilities of a device to connect to other devices, are subsumed under the term "connectivity". In the case of IoT devices, the connection is achieved using several different communication standards, like Ethernet (IEEE 802.3) and Wi-Fi (IEEE 802.11) [41], which are developed by the Institute of Electrical and Electronics Engineers (IEEE). Additionally industrial standards, like Bluetooth [42] and ZigBee [43], are also common in IoT environments. As connectivity standards evolve over time, which often includes security improvements, they are versioned. To improve user experience and compatibility, older versions are often still supported by the devices. This can decrease security, as older versions are more likely to contain security issues [44] [45].

IoTAG utilises a multi-part data structure to list the supported communication standards. The attributes of this structure are named like the standards, e.g., "IEEE802_11", "Bluetooth" and "ZigBee". Each attribute contains an collection of strings, which contain the supported versions and some times the encryption used, for example, in the case of IEEE

802.11. The version string can only contain alphanumeric versions. Bluetooth and ZigBee, for example, have numeric versions and Wi-Fi versions are named after their IEEE 802.11 suffix. As mentioned earlier, the collection can contain the supported encryption standards like “WEP”, “WPA”, “WPA2” and “WPS”.

13) Services: Services describe the ways other clients can communicate with the device. This communication also needs to be encrypted. Additionally, the software running on the device, which exposes the server to the network could have security flaws, hence the currently running version should also be included.

The services of a device are listed under a separate data structure as a collection of several services, which contain the following attributes: The name of the service, the port that is utilised by it, the protocol used for communication and the name and version of the software. The name and version of the software are combined into one string in the format <designation>-<version>. The port is a string which is the combination of the port and either TCP or UDP and is separated by a slash: <Port>/<UDP|TCP>.

B. Serialization

Now, as the data contained in IoTAG has been defined, there is a need for a uniform format to transport and process this data. The goal is to avoid incompatibilities due to misinterpretations.

For serialization, the Javascript Object Notation (JSON), according to the specification in ECMA-404 [46] and RFC 8259 [47] with UTF-8 encoding is chosen.

Because of its lower memory consumption and better computing performance, JSON is preferred to the Extensible Markup Language (XML) [48].

Below is a fully serialized IoTAG data set whose attribute names have been transferred into a uniform format. The value of the attribute ’ID’ had to be wrapped into several lines to be displayed completely.

```
{
 "Manufacturer": "Beispiel GmbH",
 "Name": "Example-Device",
 "SerialNumber": "D1.0",
 "Type": "example device",
 "ID": "2071c7736acd16f6cea3727d
        3b7ecde53f4c2e97b421f355
        0248e19d7309c636",
 "Category": "infrastructure",
 "SecureBoot": false,
 "Firmware": {
    "Version": "1.0",
    "URL": "https://192.168.102.94:10000/FirmwareInfo"
 },
 "ClientSoftware": {
    "Version": "",
    "URL": ""
 },
 "Updates": {
    "AutomaticUpdates": false,
    "AutomaticUpdatesPossible": false,
    "LastUpdateOn": "2020-08-01T00:00:00"
    "EndOfLife": "2021-01-01T00:00:00"
 },
 "Cryptography": {
    "Software": {
        "IoTAGKey": true,
        "KeyStore": true,
        "Algorithms": [
            "RSASSA-PSS",
            "SHA-256",
            "TLS_AES_128_GCM_SHA256",
            "TLS_AES_256_GCM_SHA384",
            "TLS_CHACHA20_POLY1305_SHA256",
            "aes128-ctr",
            "aes192-ctr",
            "aes256-ctr",
            "ecdsa-sha2-nistp256",
            "ecdsa-sha2-nistp384",
            "ecdsa-sha2-nistp521",
            "ssh-rsa",
            "ssh-dss",
            "ecdh-sha2-nistp256",
            "ecdh-sha2-nistp384",
            "ecdh-sha2-nistp521",
            "diffie-hellman-group-exchange-sha256",
            "hmac-sha2-256,hmac-sha2-512"
        ]
    },
    "Hardware": {
        "IoTAGKey": false,
        "KeyStore": false,
        "Algorithms": []
    }
 },
 "Connectivity": {
    "IEEE802_11": [
        "WPA2",
        "b",
        "g",
        "n",
        "ac"
    ],
    "Bluetooth": [
        "4.2"
    ],
    "ZigBee": []
 },
 "Services": [
    {
        "Name": "IoTAG",
        "Port": "27795/TCP",
        "Protocol": "HTTP/2",
        "Software": "IoTAG-Server-1.0"
    },
    {
        "Name": "SSH",
        "Port": "22/TCP",
        "Protocol": "SSH-2",
        "Software": "OpenSSH-8.1"
    }
 ]
}
```

C. Integrity

The definitions presented so far do not yet include a procedure for verifying the provided data. It is not possible to verify whether a received data set was actually provided by the device it describes. In the course of this chapter, the signature mechanism for IoTAG will be introduced. First, the signature procedure is presented, then, the generation of the data to be signed is explained and finally the complete signing process and the subsequent validation of the signature based on examples is illustrated.

1) Signature algorithm and authentication: The RSA procedure serves as the basis for the signature mechanism of IoTAG. This is an asymmetric encryption method in which a message is encrypted with the recipient’s public key, whereby the plaintext can only be restored with the corresponding private key. By reversing this procedure and encrypting a message with the sender’s private key, the source text can be calculated using its public key. This ensures that the message is only created and sent by an instance that has the private key [49]. The keys themselves are random bit sequences for which a minimum length of 2048 bits is recommended [50].

Since the RSA algorithm would always generate the same encryption text for identical messages, methods were developed that combine the plaintext with a random value, the padding, before each encryption process. The Public-Key Cryptography Standards (PKCS) define two signing procedures for RSA in PKCS #1 that take such padding into account. These procedures, called signature schemes with appendix (SSA), are RSASSA-PKCS1-v1_5 and RSASSA-PSS. The latter is preferred for new developments, which is why it is used for IoTAG signatures using the standard options defined in PKCS #1 [51].

To verify the signature, the message recipient must know the sender's public key. However, this must also ensure that an attacker has not published his key to the recipient and is therefore able to generate misleading messages whose signature is considered valid by the recipient. To counteract this, the signer's public key is published in conjunction with a certificate, which in turn is signed by a trustworthy third party and provides certainty about the origin of the verification key [49]. In IoTAG certificates are used according to the specification in ITU-T X.509 [52] and RFC 2459 [53], as they are also used in the Transport Layer Security (TLS) protocol [54].

Such a certificate can be issued directly by the manufacturer of a device and stored on the device, or it can be created when the device is set up and then signed by a local or external certification authority. In all cases it must be ensured that each device receives an individual certificate. It is the responsibility of the message recipient to check the validity of the certificate.

2) Signed dataset: After a suitable signature procedure has been selected, it is now necessary to determine which data is to be signed. Basically, the target of the signature is always the IoTAG data record in serialized form and thus a UTF-8 encoded character string. However, not this entire string is used for the signature, but instead a hash sum is calculated from it, which is then signed. The SHA-256 algorithm is used to generate this sum, as recommended by NIST (National Institute of Standards and Technology) [35].

Before the hash algorithm can be applied, the IoTAG string is converted into a byte array. Only from this array, the hash sum is calculated, to which the signature algorithm is then applied. If the array contains a terminating null byte, this is ignored in the hash calculation.

3) The signing process based on examples: This example is intended to illustrate the following sequences of the signing process: The creation of the hash sum, the signing of the hash sum and the validation of the signature. To do this, an RSA key pair and an IoTAG must first be defined. A size of 2048 bits is chosen for the key pair. For reasons of clarity, the IoTAG is not serialised in its entirety, but only using the fields "Manufacturer", "Name", "SerialNumber" and "ID". Also, no certificates, but only the required keys are used. The implementation of the program code required for the example is done in the programming language Go [55].

Before the actual signing process can be started, an RSA key pair with a size of 2048 bits and an IoTAG object with exemplary attribute values must be created:

```
privkey, _ := rsa.GenerateKey(rand.Reader, 2048)
pubkey := privkey.Public().(*rsa.PublicKey)

iotag := struct {
    Manufacturer string
    Name         string
    SerialNumber string
    ID           string
}{
    "Example Company",
    "smoke detector",
    "R1.234",
    "db0fb9870ffc08ccc" +
    "b59b9d65a0ceb0cd0" +
    "108265471a89e3c35" +
    "e21edfe7c00d3",
}
```

The IoTAG object can now be converted into a JSON object:

```
serialized, _ := json.Marshal(iotag)
```

In case of Go, the serialization process returns a non zero terminated byte array, which can be directly used for the calculation of the hash sum. The byte chain, generated by the serialization, can now be transferred to the hash algorithm:

```
hashed := sha256.Sum256(serialized)
```

By which the following Hashsumme in hexadecimal representation results:

```
f278178e0a885a074f7bf8e06968f11b
53931a00108dd46eb4b1a238dd312959
```

This can now be used to create the signature using the RSASSA-PSS procedure, which additionally requires the private RSA key:

```
signature, _ := rsa.SignPSS(rand.Reader, privkey,
    crypto.SHA256, hashed[:], nil)
```

The signature is now ready to be transmitted.

To be able to check whether the signature generated in the previous step is valid, the receiver needs the following additional information:

- The serialized IoTAG object
- The public key

In this example, it is assumed that this information has already been transmitted to the verifier of the signature and the hash operation has been performed, so that the signature verification can be executed with the corresponding parameters:

```
result := rsa.VerifyPSS(pubkey, crypto.SHA256, hashed
    [:], signature, nil)
if result == nil {
    fmt.Println("Signature valid!")
} else {
    fmt.Println("Signature invalid!")
}
```

D. Communication

The last open point to be defined, is the IoTAG related communication behaviour. This includes the retrieval of IoTAG data from a device, as well as the retrieval of software resources via a URL, which must be provided by the device

firmware via IoTAG. The same technologies are used for both procedures, which is why a general description of the communication endpoint, the transmission protocol and the data format is given, before the two procedures are explained in more detail.

1) General description: HTTP Version 2 with Transport Layer Security (TLS) is selected as the transmission protocol [56] (Hypertext Transfer Protocol Secure, HTTPS for short). For querying information, an HTTPS-capable server application must be provided as the communication endpoint, which has a trustworthy certificate for encrypted communication. This application does not have to support the full scope of the operations defined in RFC 2616 [57], but only has to be able to respond to an individual GET request by providing the respective data record. The addressed resource is determined by the respective URL.

The JSON format is used to format the data for transmission within HTTP packets.

2) Retrieving Software Resources: It was determined that the IoTAG data set provided by a device should contain a URL to obtain the latest available device firmware and, if necessary, software for client systems. It is not possible to download the software directly via this URL. Instead, it is used to perform the HTTP request described before. The response to this request contains a JSON object, which in turn has the string attributes “URL” and “Version”. This URL can now be used directly to download the firmware. The second specification informs about the version of the software.

3) Retrieving IoTAG: Every IoTAG compatible device must provide a communication interface to retrieve the IoTAG dataset. In order to make this procedure uniform, a unique HTTP URL must be defined, which is used to access a corresponding resource. This requires a uniform port number and a predefined path for the request to the HTTP server. 27795 is specified as the network port. The path consists of a single segment called “iotag”. This results in the following URL scheme, where the “<host>” specification is to be interpreted according to the definition in RFC 3986 paragraph 3.2.2:

```
https://<host>:27795/iotag
```

The example created in the course of the description of the signature process shows that in addition to the actual IoTAG data record, additional information is required to verify its correctness. This is a certificate that contains the key needed to verify the signature, as well as the signature itself. A separate JSON object is also defined for this purpose, which contains this information in the form of the attributes “IoTAG”, “Certificates” and “Signature”.

Since, the signature is present as a byte sequence, it will first be encoded to base64, which allows it to be integrated into the JSON object as a string. The format in which the certificate is stored on the respective devices depends on the implementation by the manufacturer. It must therefore be converted into a uniform format for transmission. For the transmission of ITU-T X.509 certificates in non-binary form, the encoding according to RFC 7468 [58] is suitable. Basically, the certificate is first converted into a binary structure, taking into account the encoding rules specified in ITU-T X.690 [59], and then encoded to base64, which means it can also be embedded as a string in the JSON object.

If additional certificates are required to verify the certificate, all certificates are first encoded and the resulting character strings are then concatenated. The order according to the specification in RFC 5246 chapter 7.4.2 [54] must be taken into account.

The IoTAG data record could be entered directly as an object, since it is JSON-serialized for transmission anyway. In order to check the signature, the recipient must extract the IoTAG object from the parent object. This can be done in two ways: the recipient can continue to treat the transmitted data as a character string and try to extract the IoTAG object by manipulating it. However, this procedure is unusual and involves additional development effort, since the corresponding extraction routine must be implemented. Alternatively, the received JSON object can be deserialized to an object of the respective programming language used and then be processed further.

Although, the latter approach is preferable, it also makes signature verification more problematic. To perform this step, the IoTAG object must be serialized back to a string after extraction to calculate the hash sum. This serialization produces different results depending on the software used, which ultimately results in different hash values. The problem of the different serializations can be illustrated with an example. First, an object is created in the programming language Python [60], which is identical to the object before. This object is then serialized and hashed:

```
iotag = {
    "Manufacturer": "Beispiel GmbH",
    "Name": "Rauchmelder",
    "SerialNumber": "R1.234",
    "ID":  "db0fb9870ffc08cccb59b9d"
           "65a0ceb0cd0108265471a89"
           "e3c35e21edfe7c00d3"
}
serialized = json.dumps(iotag).encode('utf-8')
hash = hashlib.sha256(serialized).hexdigest()
print(hash)
```

This process results in the following hash value:

```
5063aec9e300b6d4a61ce3dd6f7b0b42
98ddc230914ca3b5676df694fbe632e7
```

By comparing this result with the one before, it can be seen that the values are different. A signature verification based on the respective hashes would thus fail, although the information would remain unchanged.

To counter this problem, the IoTAG data set must be transferred within a JSON object in such a way that it can be extracted by deserialization without affecting the formatting. This can be achieved by treating the serialized IoTAG data for transfer as a string rather than an object. However, all JSON control characters within this string must be replaced with appropriate escape sequences before transmission to allow for

error-free interpretation. These must also be removed by the receiver before the hash calculation.

In order to avoid the resulting programming effort, a further approach is preferred. The transmission of the IoTAG data as a string is retained, but the character string resulting from its serialization is first base64 encoded. The result of this process is then set as the value of the IoTAG attribute. This enables the recipient of the data to parse the received JSON object and decode the information inside. This information will then be available in the same format as it was processed by the sender.

VIII. Conclusion

The scoring of the network is in the early stages of research and continuously being developed. There are still some points which are unclear and the individual weightings have to be adjusted in detail. Nevertheless, we show an advanced approach which can already be tested in practice.

As mentioned in our previous publication [1], we want to improve our scoring system by scanning vulnerability databases to change the scoring and warn the user, if a new vulnerability emerges.

To get the best results and an accurate overview of the devices, we proposed our standard IoTAG. It solves the problems with device detection and provides reliable information about the current security status of the individual devices.

The definition of the individual points of IoTAG is already far-reaching, but can be flexibly extended by further parameters. This should keep open the possibility to add further features (e.g., the functions of a device) or, to merge with other existing approaches.

In this paper, we have additionally shown that it is possible to implement IoTAG in Go with little effort. The same is true for the C programming language, which attests to a broad application, especially in the field of IoT. So far there are still missing further implementations and public code repositories, which we plan to submit in the near future.

In addition to the advantages of IoTAG, the view of an attacker should be briefly considered. IoTAG enables the possibility for an attacker to get all the device data from a network without much effort. This can help to identify the most vulnerable device within the network. To avoid those attacks, devices can release the IoTAG data to client systems only if their TLS certificate is signed by a trusted certification authority. Another possibility would be to store the device that is authorized to retrieve IoTAG data in the configuration.

But there is still an unsolved problem. Manufacturers must integrate IoTAG into their devices to enable the comprehensive device detection and the associated network scoring.

References

[1] S. Fischer, K. Neubauer, L. Hinterberger, B. Weber, and R. Hackenberg, "IoTAG: An Open Standard for IoT Device IdentificAtion and RecoGnition," in SECURWARE 2019, Thirteenth International Conference on Emerging Security Information, Systems and Technologies, 2019, pp. 107-113.

[2] "Rash of in-the-wild attacks permanently destroys poorly secured IoT devices," 2017, URL: https://arstechnica.com/information-technology/2017/04/rash-of-in-the-wild-attacks-permanently-destroys-poorly-secured-iot-devices/ [accessed: 2020-08-18].

[3] "Five nightmarish attacks that show the risks of IoT security," 2017, URL: https://www.zdnet.com/article/5-nightmarish-attacks-that-show-the-risks-of-iot-security/ [accessed: 2020-08-18].

[4] "Nmap: the Network Mapper - Free Security Scanner," URL: https://nmap.org [accessed: 2020-08-18].

[5] "Fing - IoT device intelligence for the connected world," URL: https://www.fing.com [accessed: 2020-08-18].

[6] M. Miettinen et al., "IOT SENTINEL Demo: Automated Device-Type Identification for Security Enforcement in IoT," in IEEE 37th International Conference on Distributed Computing Systems (ICDCS), 2017, pp. 2511-2514.

[7] T. D. Nguyen et al., "DÏOT: A Federated Self-learning Anomaly Detection System for IoT," in IEEE 39th International Conference on Distributed Computing Systems (ICDCS), 2019, pp. 756-767.

[8] T. Kohno, A. Broido, and K. C. Claffy, "Remote physical device fingerprinting," in IEEE Trans. Dependable Secure Comput., vol. 2, no. 2, April 2005, pp. 93–108.

[9] J. Cache, "Fingerprinting 802.11 implementations via statistical analysis of the duration field," Uninformed, org 5, 2006.

[10] J. Franklin, D. McCoy, P. Tabriz, V. Neagoe, J. Van Randwyk, and D. Sicker, "Passive data link layer 802.11 wireless device driver fingerprinting," in USENIX Security Symposium, USENIX, 2006, pp. 167-178.

[11] V. Brik, S. Banerjee, M. Gruteser, and S. Oh, "Wireless device identification with radiometric signatures," in International Conference on Mobile Computing and Networking, ACM, 2008, pp. 116–127.

[12] A. E. Khaled, H. Abdelsalam, L. Wyatt, and L. Choonhwa, "IoT-DDL device description language for the T in IoT," in IEEE Access 6, 2018, pp. 24048-24063.

[13] S. Kaebisch and A. Darko, "Thing description as enabler of semantic interoperability on the Web of Things," in IoT Semantic Interoperability Workshop, 2016, pp. 1-3.

[14] "Web of Things (WoT) Thing Description," Apr. 2018, URL: https://www.w3.org/TR/wot-thing-description/ [accessed: 2020-08-18].

[15] "andypitcher/IoT_Sentinel: IoT SENTINEL : Automated Device-Type Identification for Security Enforcement in IoT," December 9, 2018, URL: https://github.com/andypitcher/IoT_Sentinel [accessed: 2020-08-18].

[16] F. Loiy, A. Sivanathany, H. H. Gharakheiliy, A. Radford, and V. Sivaraman, "Systematically Evaluating Security and Privacy for Consumer IoT Devices," in IoT S&P 2017, 2017, pp. 1-6.

[17] K. C. Park and D. Shin, "Security assessment framework for IoT service," in Telecommun Syst, 2017, pp. 193–209.

[18] B. Ali and A. I. Awad, "Cyber and Physical Security Vulnerability Assessment for IoT-Based Smart Homes," in sensors journal, vol 18(3), 2018, pp. 817.

[19] R. I. Bonilla, J. Crow, L. Basantes, and L. Cruz, "A Metric for Measuring IoT Devices Security Levels," in IEEE 15th Intl Conf on Dependable, Autonomic and Secure Computing, 2017, pp. 704-709.

[20] "Norton Core Router — Secure WiFi Router," URL: https://us.norton.com/core [accessed: 2020-08-18].

[21] "Avira SafeThings™ - IoT Security for the Connected Home," URL: https://safethings.avira.com/for-partners [accessed: 2020-08-18].

[22] "Norton Core Security Score," Mar. 30, 2020, URL: https://support.norton.com/sp/en/za/norton-core-security/current/solutions/v118380521 [accessed: 2020-08-18].

[23] "How do I purchase Norton Core?," Apr. 23, 2020, URL: https://support.norton.com/sp/en/us/norton-core-security/current/solutions/v131932667 [accessed: 2020-08-18].

[24] "Service Name and Transport Protocol Port Number Registry," Aug. 12, 2020, URL: https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt [accessed: 2020-08-18].

[25] "Cryptography in IT - recommendations on encryption and procedures, Kryptographie in der IT - Empfehlungen zu Verschluesselung und Verfahren," June 17, 2016, URL: https://www.heise.de/security/artikel/Kryptographie-in-der-IT-Empfehlungen-zu-Verschluesselung-und-Verfahren-3221002.html [accessed: 2020-08-18].

[26] "CVE - Common Vulnerabilities and Exposures (CVE)," URL: https://cve.mitre.org/ [accessed: 2020-08-18].

[27] "Dienst- und Versionserkennung," URL: https://nmap.org/man/de/man-version-detection.html [accessed: 2020-08-18].
[28] "tshark - Dump and analyze network traffic," URL: https://www.wireshark.org/docs/man-pages/tshark.html [accessed: 2020-08-18].
[29] "GitHub - vanhauser-thc/thc-hydra: hydra," URL: https://github.com/vanhauser-thc/thc-hydra [accessed: 2020-08-18].
[30] "Login to the FRITZ!Box Web Interface," 2018, URL: https://avm.de/fileadmin/user_upload/Global/Service/Schnittstellen /Session-ID_english_13Nov18.pdf [accessed: 2020-08-18].
[31] "OS Detection - Nmap Network Scanning," URL: https://nmap.org/book/man-os-detection.html [accessed: 2020-08-18].
[32] L. Hinterberger, S. Fischer, B. Weber, K. Neubauer, and R. Hackenberg, "IoT Device IdentificAtion and RecoGnition (IoTAG)," in CLOUD COMPUTING 2020, The Eleventh International Conference on Cloud Computing, GRIDs, and Virtualization, Accepted, 2020.
[33] U.S. Department of Commerce and National Institute of Standards and Technology, "Secure Hash Standard (SHS)," 2015.
[34] "RFC 4648 - The Base16, Base32, and Base64 Data Encodings," Oct. 2006, URL: https://tools.ietf.org/html/rfc4648 [accessed: 2020-08-18].
[35] "NIST Policy on Hash Functions," June 22, 2020, URL: https://csrc.nist.gov/Projects/Hash-Functions/NIST-Policy-on-Hash-Functions [accessed: 2020-08-18].
[36] R. K. Dahal, J. Bhatta, and T. N. Dhamala, "Performance Analysis of SHA-2 and SHA-3 finalists," in International Journal on Cryptography and Information Security (IJCIS), Sept. 2013, pp. 720-730.
[37] U.S. Department of Commerce and National Institute of Standards and Technology, "SHA-3 Standard: Permutation-Based Hash and Extendable-Output Functions," 2015.
[38] "RFC 3986 - Uniform Resource Identifier (URI): Generic Syntax," Jan. 2005, URL: https://tools.ietf.org/html/rfc3986 [accessed: 2020-08-18].
[39] International Organization for Standardization, "Data elements and interchange formats — Information interchange — Representation of dates and times," 2004.
[40] "RFC 5656 - Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer," Dec. 2009, URL: https://tools.ietf.org/html/rfc5656 [accessed: 2020-08-18].
[41] "Institute of Electrical and Electronics Engineers," URL: https://ieeexplore.ieee.org/browse/standards/get-program/page/series?id=68 [accessed: 2020-08-18].
[42] Bluetooth SIG, Inc., "Bluetooth Core Specification, Revision 5.2," 2019.
[43] ZigBee Alliance, "ZigBee Specification," 2015.
[44] P. Kraft and A. Weyert, "Network Hacking," Franzis Verlag GmbH, 2015, pp. 345-360.
[45] J. Erickson, "Hacking," dpunkt.verlag GmbH, 2009, pp. 472-488.
[46] ECMA International, "The JSON Data Interchange Syntax," 2017.
[47] "RFC 8259 - The JavaScript Object Notation (JSON) Data Interchange Format," Dec. 2017, URL: https://tools.ietf.org/html/rfc8259 [accessed: 2020-08-18].
[48] N. Nurseitov, M. Paulson, R. Reynolds, and C. Izurieta, "Comparison of JSON and XML data interchange formats: A case study," in International Conference on Computer Applications in Industry and Engineering, CAINE, 2009, pp. 157-162.
[49] A. S. Tanenbaum, "Moderne Betriebssysteme," Pearson Deutschland GmbH, 2009, pp. 717-721.
[50] U.S. Department of Commerce and National Institute of Standards and Technology, "Recommendation for Key Management," 2015.
[51] "RFC 8017 - PKCS #1: RSA Cryptography Specifications Version 2.2," Nov. 2016, URL: https://tools.ietf.org/html/rfc8017 [accessed: 2020-08-18].
[52] International Telecommunication Union, "Recommendation ITU-T X.509," 2016.
[53] "RFC 2459 - Internet X.509 Public Key Infrastructure Certificate and CRL Profile," Jan. 1999, URL: https://tools.ietf.org/html/rfc2459 [accessed: 2020-08-18].
[54] "RFC 5246 - The Transport Layer Security (TLS) Protocol Version 1.2," Aug. 2008, URL: https://tools.ietf.org/html/rfc5246 [accessed: 2020-08-18].
[55] "The Go Programming Language," URL: https://golang.org/ [accessed: 2020-08-18].
[56] "RFC 7540 - Hypertext Transfer Protocol Version 2 (HTTP/2)," May 2015, URL: https://tools.ietf.org/html/rfc7540 [accessed: 2020-08-18].
[57] "RFC 2616 - Hypertext Transfer Protocol – HTTP/1.1," June 1999, URL: https://tools.ietf.org/html/rfc2616 [accessed: 2020-08-18].
[58] "RFC 7468 - Textual Encodings of PKIX, PKCS, and CMS Structures," Apr. 2015, URL: https://tools.ietf.org/html/rfc7468 [accessed: 2020-08-18].
[59] International Telecommunication Union, "Recommendation ITU-T X.690," 2015.
[60] "Welcome to Python.org," URL: https://www.python.org/ [accessed: 2020-08-18].

Analyzing Model Element Labels of BPMN Diagrams Provided on the Web

Christian Kop
University of Klagenfurt
Klagenfurt, Austria
e-mail: christian.kop@aau.at

Abstract— **The Business Process Model and Notation (BPMN) is the de-facto standard for process modeling. It provides four types of diagrams to cover different aspects of process modeling, ranging from process specifications itself to the specification of the interactions between the involved participants at different level of abstractions. These different types of Diagrams are Process Diagram, Collaboration Diagram, Choreography Diagram and Conversation Diagram. For all BPMN models of any of these diagram types, it is important that they are understandable to all stakeholders. The Web provides many examples of these different diagrams types. Enterprises and consultants, who offer technical solutions (i.e., BPMN tools) or consulting services for BPMN, provide these examples. Since such models are provided on the Web as introductory learning examples, such examples can also influence novice BPMN modelers. Therefore, it is worth to examine if such examples have the same quality standards as suggested in the literature. This paper, therefore, focuses on the analysis of such BPMN examples. Particularly, it focuses on the labels of model elements, since these labels represent the relationship between a BPMN model and a certain domain. Hence, this paper shows results of the analysis of model element labels that appear in Process Diagrams Collaboration Diagrams, Choreography Diagram and Conversation Diagrams.**

Keywords-Business Process Model and Notation (BPMN); Labels of Model Elements; Collaboration Diagrams; Choreography Diagrams; Conversation Diagrams

I. INTRODUCTION

As an extension to [1], this paper discusses the labels of additional Business Process Model and Notation diagrams and their model elements. The Business Process Model and Notation (BPMN) is the de-facto modeling language standard for documenting processes. For the list of model elements of BPMN, see e.g., the BPMN poster on the Web [2]. In BPMN, four different diagram types exist [3]. The most used type of diagram is the Process Diagram. It specifies the details of a single process. The other types of diagrams are Collaboration Diagram, Choreography Diagram and Conversation Diagram. With these four diagrams, a modeler can model different aspects. This ranges from the modeling of interaction of participating systems to the detailed description of the process within one of the participating systems. Even the Process Diagram itself is intended for both high-level organizational processes and lower level processes that a workflow engine can execute. A good analysis and documentation is necessary in order to understand the internal behavior of a system like an enterprise, its interaction to other participants (e.g,, customers or suppliers) and to implement process automation well.

For a good documentation of all aspect of process modeling, which is understandable by all stakeholders, skills in modeling with BPMN are very relevant. Today, reading books about BPMN or visiting BPMN courses are not the only ways to obtain these skills. Instead, it is often much easier and cheaper to click through the Web, looking and reading the diagrams, as well as the enclosed explanations. Thus, Web examples can be taken as surrogates for examples in professional literature (e.g., specialist books). Actually, the BPMN and Business Process Management (BPM) community (e.g., tool providers and consultants) also have the aim to present BPMN examples on the Web to give either an introduction of the tool features for BPMN modeling or to show modelers how these diagrams look like and how they should be modelled. Hence, a look at such Web examples and their quality for being a standard for novice modelers can be useful.

There are different aspects of how modeling quality can be defined (e.g., syntactical correctness; adequate drawing of models; adequate color and shape of model elements; adequate labels of model elements, etc.).

This work focuses on the labels of model elements. Whereas the previous work [1] focused on important model elements of Process Diagrams only, this extended version also considers the labeling styles of the three other types of BPMN diagrams. Labels on model elements (e.g., "send application" as a label example of a BPMN Activity) are important since they relate the model to the observed reality. They represent the semantic bearing parts of a domain giving the model elements and thus the whole BPMN model a certain meaning in a specific domain. Therefore, if the labels are not well chosen, a model can be more confusing than understandable and this can lead to a wrong interpretation of models. Unfortunately, if modeling tools would analyze such labels, they will not be able to give exact results if a label is correct or not. The reason is obvious. Natural language labels do not follow those strict syntactic patterns like the model elements in an artificial modeling language like BPMN. Furthermore, there are many natural languages. A certain syntactic pattern that makes up a good labeling style in English must not necessarily be applicable in another language. Therefore, tools can only make suggestions. However, if these tool suggestions do not fit with introductory learning examples (e.g., taken from the Web), they will be worth for nothing. Hence, such learning

examples are still important and the labeling style therefore has to be carefully chosen by the creators of such examples.

While labeling guidelines already exist in literature for the important model elements of Process Diagrams, no quality guidelines exist for the other three types of diagrams. However, the model elements introduced in these diagrams have a purpose. Therefore, it can be assumed that this restricts and determines the way these elements are labeled. Additionally, these new model elements are derived from model elements, which already exist in Process Diagrams. In this latter case, it can be assumed that the kind of label is oriented on the label of the model element, from which this new element is derived.

The goals of this work, therefore, are twofold. For Process Models the goals are the following:

- Check if the introductory learning examples for Process Models provided on the Web by BPMN experts (e.g., enterprises that offer BPMN tools and consultants offering consulting services) follow the label quality guidelines mentioned in literature.
- Examine if in these examples, the labels are at least well chosen. That means: Even if the labels do not exactly match the guidelines, nevertheless, they make sense in a specific context. In order to answer this, the analysis of the examples on the Web has been done on a sample extracted from the Web.

For Collaboration Diagrams, Choreography Diagrams and Conversation Diagram, the goals of this paper are the following:

- Check if the new model elements introduced in these three additional diagrams follow the assumptions mentioned above regarding to their labeling styles.
- Check if the labeling styles of model elements, which both can be modelled in Process Diagrams and the other three diagrams stay the same.

Therefore, the paper is structured as follows. In Section II, an overview of related work is given and labeling styles together with literature recommendations of good labeling styles are presented. Section III describes the preparation of the sample of Web examples for this work. Section IV focuses on the labels of model elements for BPMN Process Diagrams. It describes, which kinds of labels are used and compares these labels with labeling style recommendations in literature. Section V focuses on the three additional types of BPMN diagrams (Collaboration Diagram, Choreography Diagram and Conversation Diagram). It discusses the labels used in the model elements of these diagrams. The paper is summarized in Section VI.

II. Related Work

All aspects of the quality of process models are in the focus of the research community. In [4] the visual notations of model elements in any conceptual modeling language are examined. The author discusses the influence of this visual notation on the good or bad readability of conceptual models. Issues of deficiencies in BPMN are stated in [5] and [6]. In [7] and [8], the authors focus on the quality of BPMN models. A literature survey about business process modeling quality is given in [9]. Seven guidelines for process modeling are proposed and verified with user studies in [10]. The research in [11] focuses on the modeling language part of BPMN for describing Choreographies. The authors introduce a quality framework for checking the quality of this BPMN language part.

Some researchers have thought about automating the labeling process of business process modeling and aggregation of process models to support the comprehension of such process models [12] [13]. It was even analyzed how the style, color and arrangement of label parts on a model element improves readability [14] [15].

A. Related Work with respect to Labels of Model Elements

More detailed work on labels of BPMN model elements itself was done in [16] - [18]. These research works are based on data sets of process models from industry. Good labeling styles of Activities, Events and Gateways for three different natural languages were proposed and recommended in [16]. There, violations of these labeling styles are described. Table I gives an overview of the labeling styles, which will be discussed in detail afterwards.

TABLE I: Overview of Labeling styles

Model element category	**Labeling style**
Activity	• Verb Object Style • Action Noun Style • Descriptive
Gateway	• Question with Noun and Verb in Past Participle • Infinitive Verb Question • Object with Adjective Question • Equation Question
Event	• Verb in Past Participle Style • Predicative Adjective Style • Categorization Style

Activities subsume Sub Processes, Tasks and Call Activities. In all cases, the working step within a process, are described. For the labels of Activities, the following styles were found in this literature:

- *Verb Object Style*: A label that starts with a verb expressing the activity followed by an object, on which this activity is executed (e.g., “create document”).
- *Action Noun Style*: This style has three sub styles: a) A label that has either a nominalized verb only or a compound noun consisting of a verb as the head of this compound noun (e.g., “creation”, “document creation”). b) The Noun can also be a noun phrase with the preposition “of” in between (e.g., “creation of document”). c) Finally, the *Action Noun Style* can also start with a gerund followed by a noun (e.g., “creating document”).

- The style called *Descriptive* is a style consisting of a subject, a verb in third person singular and an object (e.g., "author writes book").

Beside this, there are also labels that do not follow a good style at all. These are labels with nouns only and no verbs at all (e.g., "error"). According to literature [17], the *Verb Object Style* is the most recommended style that should be used for modeling Activities.

With Gateways, a workflow can be divided into several paths, but different paths can also be merged. Most recommended Gateway labeling styles in literature have in common that they should end with a question mark ("?"). Thus, the literature assumes that these kinds of styles are mainly used for Exclusive (XOR) and Inclusive (OR) Gateways since in these Gateways a decision is made, which can be expressed as a question. On contrary, a Parallel (AND) Gateway does not need such a label since no decision is made. Such questions in Gateway labels can be expressed in one of the following styles:

- *Question with Noun and Verb in Past Participle* (e.g., "document created?")
- *Infinitive Verb Question* (e.g., "approve contract?")
- *Object with Adjective Question*: A phrase consisting of an object followed by an adjective or an auxiliary and an adjective (e.g., "parts available?" or "parts are available?")
- *Equation Question*: A phrase consisting of an object followed by a logical operator and a value (e.g., "amount is greater than $ 200").

A counter example for good quality again is a noun only (e.g., "result?"), It is not possible to derive a clear decision from such a kind of label. For Gateways, the most recommended labeling style is *Question with Noun and Verb in Past Participle* [17].

Finally, events that can occur within a process are modelled with the model element Event.

Labeling styles for Events can be classified as followed:

- *Verb in Past Participle Style*: This can be characterized by an object followed by a verb in past participle or followed by a (modal) auxiliary and a verb in past participle (e.g., "document created", "document has been created", "document is created", "document must be created")
- *Predicative Adjective Style*: Here, a noun together with a predicative adjective is used to label an Event (e.g., "document correct" or "document is correct").
- *Categorization Style*: Two nouns are related with a verb (mainly the verb "is") in order to express that the term specified with the first noun can be categorized according to the term expressed with the second noun (e.g., "person is author").

Modelers also use labels that better should not be used for Events at all, since they do not provide sufficient information to a reader. For instance, they use a noun only (e.g., "inquiry"). The *Verb in Past Participle Style* is the one, which is most recommended as a labeling style for Events [17].

Beside simple labels, it has also been examined in literature that modelers use complex phrases and sentences for Activity labels instead of drawing more model elements with simpler, so called canonical labels. Especially in [18], these kinds of inconsistent use of labeling, so called non-canonical patterns, are examined. Three categories of complex, non-canonical label patterns were detected:

- *Complex control flow label*: The label of an Activity consists of a sequence of verbs, each describing an Activity, which are concatenated with "or" or "and". This verb sequence, however, implicitly expresses a decision (in the case of "or") or a parallel respectively a sequential execution of several Activities (in the case of "and"). It does not express an atomic working step. Thus, instead of one Activity with a complex label, several Activities with simpler labels together with control flows can also be used. Other complex labels of that kind are phrases, which end with "as required", "as / if needed", as well as sentences or phrases expressing an iteration (e.g., "while …. ", "repeat until … ", "for each …").
- *Extra specification of data, resources and time*: In this category, the label of the model element not only contains the necessary information, but also additional information that is often given in some sorts of brackets (e.g., "clear differences (inventory management)"). Most often, either this extra information should be itself explicitly modelled with a model element like an Event, Activity or Gateway or this extra information is useless.
- *Implicit Action and Decision*: Here, the label and the model element do not fit. For instance, the label of an Activity is expressed in terms of a pattern that is typically used for an Event (e.g., "order received" instead of "receive order").

In literature, these categories of non-canonical labels are seen as patterns that can confuse the reader of a model.

The guidelines for labeling discussed in literature are focusing on model elements for Process Models. No explicit guidelines exist for typical model elements of Collaboration Diagrams, Choreography Diagrams and Conversation Diagrams. However, the model elements of these three Diagrams have either a very specific purpose (e.g., Pools, Lanes) or the model elements of these Diagrams can be derived from model elements that already exist in Process Diagrams. If model elements have a specific purpose, then this can restrict the way in which they can be labeled. If they can be derived from model elements that already exist in Process Diagrams then also the labeling styles and guidelines of the exiting model elements can be applied to these "new" model elements (e.g., model elements of Choreography and Conversation Diagrams).

B. Focus of this Work with respect to Related Work

In this work, the labels of the model elements are also examined. For analyzing the labels of model elements in Process Diagrams, this work does not only consider Activities, Gateways and Events as such, but also explores different types of Activities, Gateways and Events in detail.

In addition to previous work in the related literature about labeling guidelines for Process Diagrams model elements, the main emphasis of this work can be characterized as follows: Instead of working with data sets from industry, the aim of this paper is to look for BPMN examples on the Web. Existing results of labeling guidelines in literature are taken as a reference. With this as a basis, the Web examples are examined and compared with the given guidelines.

Furthermore, the focus is not only on model elements of Process Diagrams as in previous literature. In addition, it also focuses on the label analysis of new model elements, which appear in Collaboration Diagrams, Choreography Diagrams and Conversation Diagrams. Once again, the Web was taken as a resource for analyzing the modelled examples of these diagrams.

III. PREPARATION OF THE WORK

In order to check how different enterprises, which sell BPMN modelling tools, as well as consultants, provide BPMN diagram examples, the following procedure was executed to get the sample.

For Process Diagrams, in the first step, the search term "BPMN" was entered into the search field of Google. This search engine was used as a means to choose the sample. In order to get diagrams first and not descriptions of BPMN, the image result list of the search engine was used. Here, it was expected to get various images of BPMN diagram examples. Once the search engine generated the list of diagrams, in the second step the list was manually examined. For each image, it was first decided if this image is really a BPMN diagram example in English provided at a Web site or if it is not. If it was indeed such a diagram, then the link to the respective Web site, from which the search engine listed the image, was collected. For this purpose, the link was entered into a file in order to generate a list of Uniform Resource Locator (URL) links. At the end of this URL collection step, a list of URL links, each containing at least one image of a BPMN diagram was collected in the file. In the third step, the file with these links was further examined. For doing this, the file with the URL link list was automatically scanned and each link was grouped to a Web domain.

In the fourth step, each link, as well as the additional link to the more general Web domain, was once again further examined. From all these sources, images of BPMN Process Diagram examples were downloaded and collected on the local file system. The aim of this fourth step was to find more BPMN images provided at this Web domain. Much more images were found and collected.

In the fifth step, all these collected images were manually examined and the individual BPMN model elements together with their features and labels for each diagram and domain were transcribed into a database. This data set was then analyzed according to the aim of this work.

In total, 43 Web domains and the BPMN 2.0 by Example document of Object Management Group (OMG) [19] were examined. The BPMN 2.0 by Example document was included, since this is also an important information resource about BPMN on the Web. During this collection phase, images, which were not readable, were filtered out. Images that are used as BPMN counter examples (i.e., how a BPMN diagram should not be modeled) were filtered out too, since the focus is on models that are seen as correct by the provider. The remaining examples, which in total are 346 diagram images of BPMN model examples were used for this work. Furthermore, only distinct labels were analyzed. This should avoid that a certain labeling style appears too often just because the same label (e.g., "order product") is used in many examples.

For Collaboration Diagrams, Choreography Diagrams and Conversation Diagrams in the first step the names of these Diagrams were explicitly entered into the search text field of the search engine in order to prevent results that show Process Diagrams only. Once again, the image result list was examined. Since there were not so many of these diagrams found on the Web, the rest of the collection and analysis procedure was done manually. Once the diagrams were collected, the labels of the model elements on each of diagrams were examined. Particularly, the labels were checked if there are any modeling preferences regarding a certain given labeling style. Once again, only distinct labels were analyzed.

IV. PROCESS DIAGRAM

The BPMN Process Diagram is the most used type of the four BPMN diagrams. A BPMN Process Diagram specifies the flow of the working steps for processes that usually appear in enterprises in order to produce a value (e.g., a product or a service). However, it also can be used to model any kinds of processes. For example, computer-supported parts of processes (workflows) can also be modeled with this diagram. Therefore, it is obvious that this type of diagram is very important. Figure 1 shows an excerpt of such a diagram. Here, the first two possible steps in the process of how to write a thesis are specified.

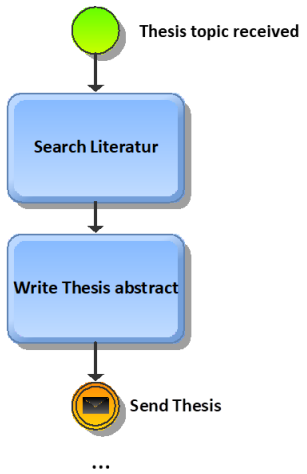


Figure 1. Except of Process Diagram

In the Subsections A to D, the analyzed distinct labels of important kinds of model elements specified in Process Diagram examples on the Web are introduced and discussed. They are discussed with respect to the recommendations in

literature mentioned in Section II. These model elements have the characteristics that

a) quality guidelines already exist in literature and

b) they appear in nearly every introductory example of a Process Diagram.

Especially, this holds for Tasks as a subset of Activity and Events. Gateways on the other hand have to be used as soon as a process model does not have only a single sequence, but the specified process in the process model branches to several paths. Thus, in most process models, except the most trivial ones, Gateways are important. Furthermore, these labels are analyzed in detail according to the specific model element, since for different model elements different labeling strategies are needed. In addition to these more important types of model elements, also model elements of minor importance will be discussed in Subsection E. However, since these types of model elements in Subsection E do not play an important role, no labeling quality guidelines exist, to which the label of these model elements can be compared.

A. Labels of Activities

As mentioned in Section II, Activities are the working parts in a process. Activities can be divided into the following categories: Task, Sub Process and Call Activity.

A Task is a single atomic working step of someone or something within the process. It is atomic since it cannot be split into smaller pieces. The OMG BPMN specification lists the following Task types: Task with no specific type (untyped Task), Send Task, Receive Task, User Task, Manual Task, Script Task, Service Task and Business Rule Task (see Figure 2 for the graphical notations).

Figure 2. Task types in BPMN

These Task types have different meanings. For instance, a User Task is a Task, where a human performs this Task with the assistance of a software application. A Manual Task is a task that is also done by a human but without any assistance of a software application. A modeler can model a task as a Send Task, if during this Task any kind of information or thing is send. S/he can model a Task as a Receive Task if any kind of information or thing is received within the process. The Script Task executes predefined scripts. In a Service Task, a predefined business logic is executed. In a Business Rule Task, predefined decisions are made. If the modeler do not want to give the modeled Task a certain semantic, then s/he models an untyped Task. After the examination of Process Diagrams, it turned out that the untyped Task was the dominating task type. About two third of all tasks were untyped Tasks. The next frequent Task was the User Task. About a fifth of all the Tasks were User Tasks. The rest were Service Tasks, Manual Tasks, Send Tasks, Receive Tasks and Script Tasks. Business Rule Tasks appear very seldom in the sample.

Sub Processes are parts in a process, which can be splitted into smaller pieces. These pieces can be itself Tasks or Sub Processes. Hence, Sub Processes represent processes within the whole process. They establish a nested hierarchy of working parts. Sub Processes can be divided into the following categories according to the OMG: Untyped Sub Process, Event Sub Process, Transaction or Ad hoc Sub Process. Figure 3 shows the graphical notations of the different types of Sub processes.

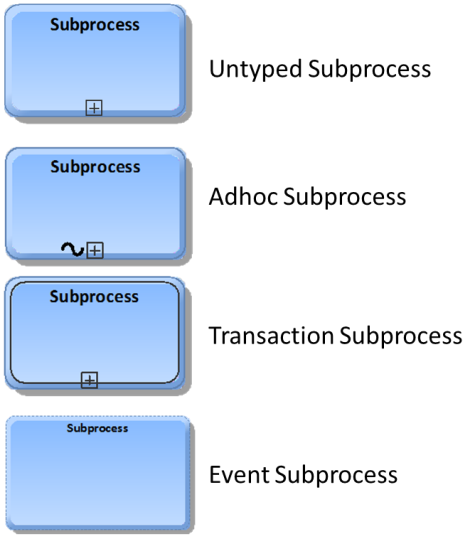


Figure 3. Sub Process types in BPMN

An Event Sub Process is a Sub Process that is triggered by Events. A Transaction Sub Process is a process that must reach a consistent state. In an Ad-hoc Sub Process the parts of it (i.e., Tasks, other Sub Processes) do not have a causal dependency on each other. They can even be executed in parallel. Once again, if the modeler do not want to specify the specific category of a Sub Process, then s/he uses the untyped Sub Process.

In the given sample, this untyped Sub Process is the category that is mostly used. More than 80 % of all modeled Sub Processes are untyped Sub Processes.

Finally, a Call Activity (see Figure 4 for the graphical notation) refers to a Sub Process that is globally specified once, instead of directly embedded into the overall process.

Figure 4. Call Activity

To summarize, Activities represent those parts of a process where somebody or something should act in order to progress the process. Therefore, an active verb, which is the best word category for acting, should be used to label these model elements. In literature, the *Verb Object Style* is preferred. An object itself can be a noun (simple ore compound) or a noun phrase.

In the sample, 944 distinct Task labels were found. The majority of these distinct Task labels (75 %), , follow this *Verb Object Style*, where the object is a noun and the direct object of the verb (e.g., "specify vacancy", "ship item", "review results"). In some cases, an article is added (e.g., "select a pizza"). Only in 2 % of all cases, a single verb or a verb together with an adverb is the only label for a Task (e.g., "publish", "rate negatively"). In 13 % of the cases, the Task labels extend the suggested *Verb Object Style* a little bit. In these labels, the object is a noun phrase (e.g., "nomination form" in "send nomination form"). In addition, cases were found, where the object is an indirect object (e.g., "communicate to customer") or there are two objects (direct and indirect object) following the verb (e.g., "deliver books to customer"). In 10 % of the Task labels, the modeler used other labeling styles for Tasks. For instance:

- They used nominalization of a verb (e.g., "delivery").
- They used full sentences (e.g., "why have you bought so many sticks of sausage?").
- They concatenated verbs (e.g., "add paperwork and move package to pick area").
- They used a condition phrase ("check if extra insurance is necessary").

To summarize, the *Verb Object Style* preferred in literature is also used in the majority of cases on the Web (see Figure 5).

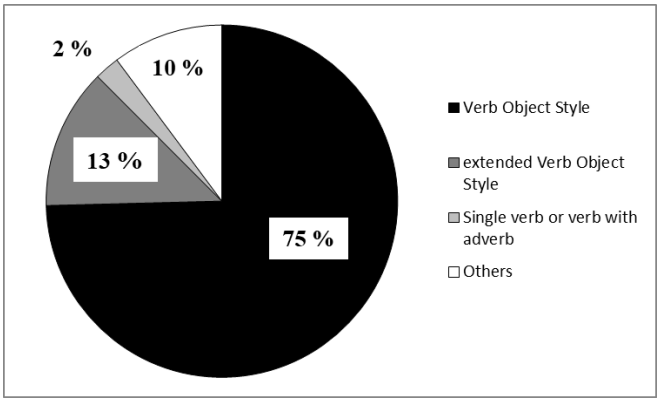


Figure 5. Percentage of Task labeling styles.

For Sub Processes, the situation is as follows: From the 85 distinct labels of normal Sub Processes, 42 % have a nominalization of the verb (e.g., "ordering", "creation") as their label (i.e., *Action Noun Style*). In 55 % of the cases, Sub Processes follow the *Verb Object Style*. Either the rest does not have a label or it is a complex sentence (e.g., "send out application forms & reminders"). Hence, no definite labeling preference can be found in these examples from the Web.

There are not enough Event Sub Processes and Transactions in the sample. Therefore, here it is hard to make a good proposition. In these few examples, the labels follow the *Verb Object Style*. There are also not so many Call Activities in order to make a proposition. Therefore, it can only be observed here that modelers prefer the *Action Noun Style* instead of the *Verb Object Style*.

B. Relationships between Labels and Specific Task Types

Since about a fifth of all modeled Tasks are modeled as User Tasks, it is interesting to see, what is modeled as a User Task. Especially, it is interesting to see, what is modeled as a User Task in comparison to what is modeled as a Manual Task. Therefore, the labels of the two Tasks are further analyzed.

From the point of view of the BPMN specification [20], there is a clear distinction between a User Task and a Manual Task. A person performs a User Task but a software application assists this person. A Manual Task is also performed by a person, but without assistance of a software application system.

It could be expected that labels for Tasks that represent a software application support differ from the labels of Manual Tasks. However, according to the labels it is not always possible to differentiate between a User Task and a Manual Task. Of course, labels with a verb were found that fit with the purpose of a User Task (e.g., "edit 1st level ticket", "fill in purchase form", "book flight", "find student's position"). On the other hand also labels were found, which do not perfectly fit with the purpose of a User Task (e.g., "hire staff", "plan interview", "read book", "rent office", "ship book", "train new employee on job specifics", "discuss nominations", "announce Nobel prices laureate", etc.). The labels for User Tasks and Manual Tasks are set arbitrarily. One interpretation can be that it is the modelers decision to see something as a Manual Task (without software application support) or a User Task (with software application support) and it depends on the purpose of the model (i.e., whether it is a workflow model or not). Particularly, a User Task can be more than a simple user interaction with the Information System. Thus, if a workflow for a workflow engine is specified with BPMN then it seems that every Manual Task can also become a User Task. A second interpretation can be that modelers of these introductory learning examples do not really want to distinguish between User Tasks and Manual Tasks at all. Therefore, they prefer to model a User Task even in a situation where a Manual Task would be the right choice.

The frequency of other task types is very low and, except for Send Tasks, no relationship between labels and these Task types were found. For the 38 distinct labels of Send Tasks, in this sample it turned out that 53 % of the distinct Send Task labels start with the verb "send". Further, 26 % have a verb like "email", "inform", "notify", "distribute", "post", "submit", "order". All these other label examples can be seen as variants of sending. Thus, it can be concluded that

labels of a Send Task are in accordance with the purpose of this Task type.

C. Labels of Events

The next important model element of BPMN is the Event. BPMN distinguishes between the following categories: Events that start a process (Start Event), Events that finish a process (End Event) and Events that can happen during process execution (Intermediate Event). There is also a special type of Event, which can be placed on the boundary of an Activity (Boundary Event). Furthermore, in each of these categories an Event can have different types. Some of these types are untyped Event, Message Event, Timer Event, Error Event and Conditional Event. If the modeler do not want to model a specific type of Event, then s/he models an untyped Event. Finally, for some of the specific Event types, it can be distinguished whether an Event is triggered (throwing Event) or an Event is received (catching Event). The semantic of a certain Event depends on the combination of the aforementioned category, type and if it is a throwing or catching Event. For instance, an Intermediate catching Message Event means that within a process, the process execution waits until the process receives a message, an information or thing. After it has been received, the process continues. A throwing Intermediate Message Event means that at a certain state in the process, a message, information or thing is sent to a recipient that has to catch this message information or thing respectively. Immediately after sending it, the process execution continues with the next process step. With a Timer Event, anything that is related to time (e.g., a certain point in time, a duration etc.) can be specified. For more explanations of the different meanings of Events, the reader is referred to the OMG BPMN specification [19] [20]. Figure 6 shows some Events. This list however is not complete since there are many other types of Events, which can occur in the combination of a Start- Intermediate- or End Event and whether they are catching or throwing. For a detailed list, the reader is referred to [2].

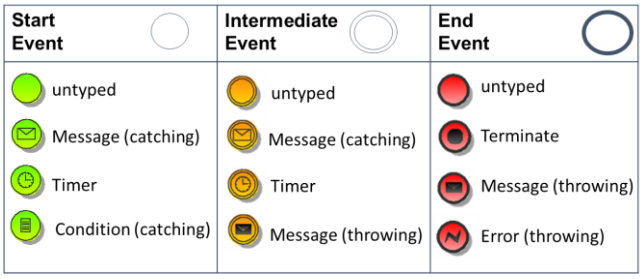


Figure 6. An excerpt of Event categories and types

When talking about labels on Events, firstly, it has to be examined if Events have labels. While BPMN modelers always give labels for Tasks, they are not so systematic if they have to specify labels for Events. From all the Start Events found in the diagrams of the sample, 46 % do not have a label. From these, most of the Events (86 %) are untyped Events (i.e., Events that are not further classified to a specific type). However, a few cases were also found with Link Events, Message Events and Signal Events that have no labels. For Intermediate Events, fewer cases with no labels exist. Only in 14 % of all Intermediate Events, no labels were detected. Particularly, the Intermediate Timer Event and Message Event are those types with no labels. These two Event types also had a high frequency within the Intermediate Events types. There are 38 % of all Boundary Events, which do not have a label. There are 56 % of all End Events, which do not have a label. From these End Events with no label, 81 % are untyped Events.

To summarize this, for the examples provided on the Web, the modelers responsible for these examples especially do not see the necessity to label Start and End Events. Particularly, this happened if these Start and End Events are untyped Events. Unlabeled model elements, however, cannot be understood well. If novice process modelers see such unlabeled model element examples on the Web, they might take it as a standard although they should avoided it.

After the examination of Events with no labels, it is interesting to see what kind of labels Events have. It is expected that specific Event types have specific types of labels. For example, Message Events and Timer Events are labelled in different ways. For this analysis, six Event types were further examined, since these Event types cover 87 % of all Event types in the sample. These Event types are: Timer Event, Message Event, Signal Event, Compensation Event, Terminate End Event and the untyped Event.

All labels of the Timer Event have, of course, in common that they specify time. However, this is done in various ways. Table II presents a list of representative Timer Event labels. In this list, the grouping of the individual labels, suggests label patterns of similar structure.

From the examples, it can be seen that they are not in accordance with the Event labeling style recommended in literature (*Verb in Past Participle Style*). Nevertheless, in the context of a Timer Event, many of these labels are appropriate.

TABLE II. TYPICAL LABELS OF TIMER EVENTS

• wait until next business day • 24h; 10 min; 60 minutes; one week; 2 weeks; 24 hours; 14 days; 48-hours • september year n-1 • wait 6 days; wait some time; wait until thursday, 9am • 1st day of month; 20th of each month • 3 business days • friday at 6 pm pacific time; friday, 6 pm pacific time	• timeout; time out (1 week); order timed out • content expired (5 days) • delay 6 days; delay 6 days from announcement • < 60 min; > 60 min • expires at set deadline • auction over • 10 min wait • 12 o'clock • start time; finish time • on next Wednesday • start on Friday • every 10 minuts; every 24 hours

For a Message Event, it has to be distinguished between a throwing Message Event and a catching Message Event. Usually, it could be expected that a catching Message Event follows the *Verb in Past Participle Style*. However, the found catching Message Events have a greater variety. Beside the typical *Verb in Past Participle Style*, also catching Message Events were found that consists of

- a noun (compound noun) or noun phrase only (e.g., "payment", "complaints to customer service")
- a complete sentence (e.g., "where is my pizza")
- a verb in past participle only (e.g., "paid").
- a noun with an adjective (e.g., "assignment complete")

Unfortunately, not so much throwing Message Events were found. Most of these throwing Message Events had no labels at all. Based on the remaining throwing Message Events with labels, it can be said that throwing Message Events use the *Verb Object Style* as it is usual in Task labels. Since a throwing Message Event can be used as an alternative for a Send Task, this labeling style is adequate. The literature recommends the *Verb in Past Participle Style* for an Event but does not consider the specific type of Event.

The found Signal Events (catching or throwing) follow the *Verb in Past Participle Style* to a large extent. Beside this, the following interesting label examples were also found: "on alert", "undeliverable". According to literature, these examples would be out of the scope of the recommendations.

Many of the Compensation Events do not have a label. Those that have a label, mainly follow the *Verb Object Style* (e.g., "cancel reservation", "undo book travel"). Many of these labelled Compensation Events are throwing Events. Once again, this is out of scope of the recommendation in the literature, which in general prefers the *Verb in Past Participle Style* for Events. However, in this special case of throwing Events, which rather express an active action than a passive reaction, the *Verb Object Style* makes sense.

Most of the Terminate End Events do not have a label. The few remaining Terminate End Events with labels follow the *Verb in Past Participle Style* or just have the label "terminate" or "end", respectively.

The analysis of untyped Events is split into the analysis of Intermediate Events, Start Events and End Events. Untyped Boundary Events with labels do not appear in the sample. This analysis provides the following results. Untyped Intermediate Events follow the *Verb in Past Participle Style*. The labels of untyped Start Events do not only follow this style. Instead, some of them only have

- a noun, compound noun or noun phrase (e.g., "application", "existing process"),
- an adjective (e.g., "hungry")
- phrases starting with an adjective (e.g., "hungry for pizza"),
- simple sentences (e.g., "the store opens").

The labels of untyped End Events follow the *Verb in Past Participle Style* largely.

D. *Labels of Gateways*

BPMN distinguishes seven types of gateways: Exclusive (XOR) Gateway, Parallel (AND) Gateway, Inclusive (OR) Gateway, Event-based Gateway, Complex Gateway and two special Gateways that should be modelled at the beginning of a process. Particularly, these are the Exclusive Instantiating Event-based Gateway and the Parallel Instantiating Event-based Gateway. For those types that appear most in the sample, the meaning will be explained. In Figure 7, the graphical notations of the most important types of Gateways are listed.

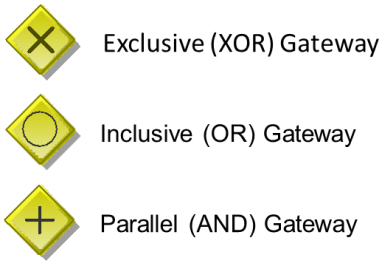


Figure 7. Most important types of Gateways

In general all gateways control, which of the several branching paths in a process are executed. The Gateways listed in Figure 7 also control how these paths can be merged together in a process. They do it however in different ways. A Parallel (AND) Gateway activates all following paths. This kind of Gateway also waits until all paths are executed if the paths are merged together. Only if all the paths are executed, the process execution after the merge continues. The Exclusive (XOR) Gateway depends on a condition expression. On the basis of the condition, it is decided, which of the several path is executed. Only one of these paths can be executed. During the merging of the paths, the Exclusive (XOR) Gateway just continues the process execution for every paths that was executed before. The Inclusive (OR) Gateway also depends on conditions but a subset of all the several paths and even all the paths can be executed. This depends on the conditions, which become true. During the merging-point the Inclusive (OR) Gateway knows the paths that were executed before and waits until all executed paths are finished before the process continues after the merge-point of an Inclusive (OR) Gateway.

With respect to the analysis of labeling, of course only the Exclusive (XOR) and Inclusive (OR) Gateways were analyzed. It is not necessary to analyze Parallel (AND) Gateways, since in these Gateways all the following branching paths are executed. Therefore, a label that specifies a condition is not necessary. In addition, none of the three Gateways (Parallel, Exclusive, and Inclusive) is analyzed at the merging-point, since they usually also do not have a label. Hence, it is only interesting how the conditions that should appear as labels on Exclusive (XOR) Gateways and Inclusive (OR) Gateways look like.

The labels of these Gateways vary. The style *Question with Noun and Verb in Past Participle* is not the only one. Again, additional patterns exist:

- nouns, compound nouns and noun phrases,
- verbs in past participle only,
- states of an object (i.e., where the state is represented by an adjective or by the word "ok"),
- comparisons with mathematical operators (e.g., ">", "<") or with words (e.g., "above").

What is common to many labels is the character "?" at the end of the label. Many Exclusive Gateways and Inclusive Gateways even do not have a label although they branch the process into two or several paths. Such cases once again can be seen as a contradiction to the recommendations in literature.

Beside the label of the Gateway itself, it is also important to analyze the labels on the Sequence Flows, which leave the Exclusive Gateways and Inclusive Gateways. About a fifth of all these labels are the words "yes" and "no", respectively. The rest varies. These variations can be seen in Table III, where some of these labels are listed.

TABLE III. LABEL EXAMPLES ON SEQUENCE FLOWS

• "1" • ">=20" • "40 % " • "yes" • "2nd level issue" • "50 % education training" • "all items available" • "allow extension" • "bicycle costs >= 500 usd" • "capacity & parts available" • "capacity not available"	• "capacity ok" • "employee is ready for work" • "fix in release" • "in stock" • "is junk mail" • "no more responses" • "not accepted" • "payment received == false" • "purchase 1" • "put on hold" • "ready with request" • "simple")

Process Diagrams intended for workflows also have Gateway labels like "${order.price <= 250}" or "${!approved}".

E. Labels of Data Object, Data Store, Text Annotation

Model elements that play a minor role in Process Diagrams are Data Object, Date Store and Text Annotation. A Data Object is any data that is processed in Activities. If data is taken from a certain storage (e.g., a file or database table), then this can be modeled with the model element Data Store. Finally, if the modeler would like to add any additional textual information to the process model, s/he can do it with the model element Text Annotation. Figure 8 shows the graphical notations of these model elements.

In labels of Data Objects also noun phrases appear. They can be complemented with verbs in participle or adjectives in order to express the state of an object (e.g., "job description", "job description [endorsed]"). Some modelers extend this verb in participle with brackets. Nouns and noun phrases are the typical labels for Data Stores. Sometimes, the abbreviation "db" or the word "database" complements the label. Since Text Annotations are just comments or additional information given by the modeler, the modeler can use any phrase or sentences to label a Text Annotation.

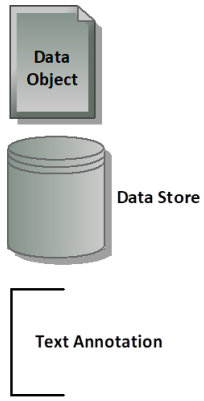


Figure 8. Data Object, Data Store and Text Annotation

V. COLLABORATION DIAGRAM, COREOGRAPHY DIAGRAM AND CONVERSATION DIAGRAM

As already mentioned in the introduction, BPMN offers three additional diagram types. These diagrams types are Collaboration Diagram, Choreography Diagram and Conversation Diagram. The following subsections describe these diagrams. First, the new model elements in these diagrams will be explained. Afterwards, the labels of these model elements will be discussed.

A. Collaboration Diagram

Whereas, a process modeler can specify the flow of activities within a single process, s/he cannot specify how two or more systems interact. Such information is modelled with a Collaboration Diagram. Therefore, the Collaboration Diagram is an extension of a process diagram. For specifying the flow of activities within each of the processes, the same model elements are used (Activity, Event, Gateway). In addition to that, the following new model elements appear in a Collaboration Diagram: Pool, Lane, and Message Flow. A Pool is a system or the role of a system in an interaction scenario that embeds a specific process. For instance, if a modeler wants to specify the surrounding system of a process, which contains this process (i.e., a department, an enterprise, a technical system or role of a system) then s/he uses the model element Pool to model it. If this system is more complex and contains subsystems then theses subsystems are modeled using the model element Lane. A typical example of a complex system can be an enterprise, which is modelled as a Pool. If it is necessary to model certain departments of that enterprise, then these departments are modeled as Lanes. In order to specify the interaction between the processes, the Pools can exchange messages. This is modelled with the model element Message Flow. Messages Flows represent simple messages, information exchanges or even material things (e.g., certain products)

Hence, a Message Flow represents anything from a simple information to a more complex domain entity like a product, contract or money that is transferred from one Pool to another. Figure 9 shows two Pools exchanging Messages. In this example, the Pools are collapsed. However, Pools in a Collaboration can also be expanded. Then each pool contains its specific process modelled with the model elements of a Process Diagram.

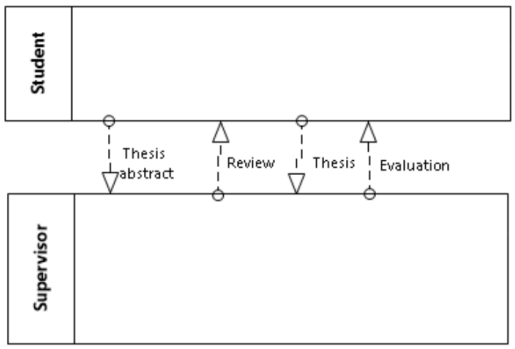


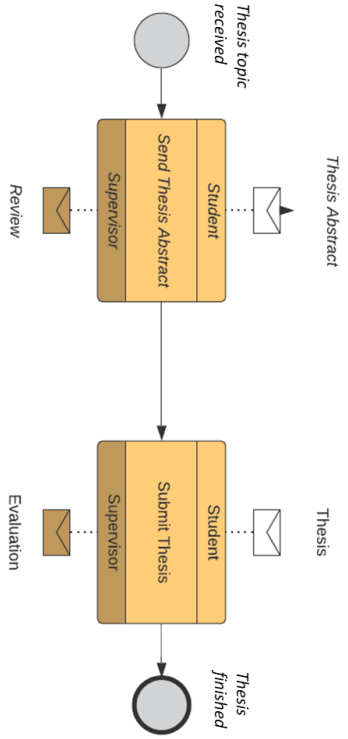
Figure 9. Collaboration Diagram with collapsed Pools

Looking at the labels of the model elements used in Collaboration Diagrams, the following can be said: The model elements that are already in use in Process Diagrams are labeled in the same ways as described in the section about Process Diagrams (Section IV). The model elements Pool, Lane and Message Flows are labeled as follows:

Nouns and noun phrases dominate the labels of Pools and Lanes. In 88 % of the cases, a label of a Pool is a noun phrase. For Lanes it is even 100 % in the sample. Typical labels on Message Flows are nouns or noun phrases only. In 69 % of the cases, a label of a Message Flow is a noun or a noun phrase. However, the labels of Message Flows can also follow other styles. Such a style for instance is *Verb in Past Participle Style*. An example for this style is e.g., "letter received". Some Message Flows follow the *Verb Object Style* (e.g., "send letter"). Message Flows can have the following labels as well: "100$", "give me 100$", "here is your medicine", "pickup your medicine and you can leave". In these special cases of whole sentences, modelers use message flows mainly to represent the concrete oral communication between persons represented by the Pools. In some Collaboration Diagrams, the Message Flows do not have a label. In these Diagrams, the modelers seem to assume that the semantic of message in the Message Flow can be derived from the involved Events.

B. Choreography Diagram

The Choreography Diagram was introduced in Version 2.0 of BPMN. A Choreography Diagram focuses on the interaction of messages between Pools. Here Pools are also called participants. It is based on a Collaboration Diagram but instead of presenting all interaction details between pools, as well as all the process details within each pool it describes the interaction in a more compact manner. Therefore, this diagram introduces the new model element: Choreography. Figure 10 shows a simple example of a Choreography Diagram. It contains two Choreographies connected by a Sequence Flow.



.

Figure 10. Choreography Diagram

The Choreography is a radiused rectangle, which consists of three sections. The outer sections represent the involved participants (Pools) in this Choreography. The section in the middle represents the activity that triggers or receives the messages. This activity can be either a Choreography Task or a Sub Choreography. Sometimes also the messages itself are presented with a letter-envelope icon related to the Choreography. A Sequence Flow can connect each Choreography to other Choreographies. Similar to Process Diagrams, such a Choreography Diagram can split into several paths. These paths can also once again merge somewhere in the diagram. Therefore, a Choreography Diagrams contains Gateways too. In addition, such a Diagram can also contain Events. At least one start Event and one End Event are mandatory.

An examination of several Choreography Diagrams found on the Web provides the following results about the labels.

The outer sections with the participants (Pools) involved in a Choreography always have a noun as a label. In the middle section where the activity is defined, the following was found. In 73 % of the cases, the section, representing the activity (Chorography Task or Sub Choreography), is labeled with the *Verb Object Style* (e.g., "confirm order"). In 10 % of the cases, the *Action Noun Style* (e.g., "order rejection") is

used. In 11 % of the cases, these labels were also nouns, noun phrases or nouns with an adjective (e.g., "insufficient credit"). In addition, a few cases (6 %) used non-canonical patterns (e.g., "pick and drop Customer") or other labeling styles (e.g., "payment ok"). Unfortunately, far too few Sub Choreographies exist in the sample. Therefore, the analysis and analysis results do not distinguish explicitly between Choreography Tasks and Sub Choreographies. However, in these rare cases of Sub Choreography examples, it was examined that the *Verb Object Style* also dominates.

In some cases, it was also interesting to see a switch of labeling style in the same diagram. The creator of this diagram used the *Verb Object Style* to a large extend but then used noun or noun phrase only. Since this middle section of the model element Choreography is derived from a Task or Sub Process, which are modelled in a Collaboration Diagram, then the *Verb Object ,Style* should be used. If it would be derived form a Sub Process, both the *Verb Object Style* and *Action Noun Style* would be appropriate. However, other kind of labeling patterns do not fit in this middle section of a Choreography.

For most of the Messages related to a Choreography, nouns or noun phrases were used as labels. In one exceptional case of a diagram, such a label is a whole sentence.

Nearly in all Choreography Diagrams, the Start Events and End Events had no label, since they just represent the trivial fact that the Choreographies have a start and an end. The few Intermediate Events that appeared in the diagram had similar variants of labeling styles like the Intermediate Events in Process Diagrams. Those kinds of Gateways, which appeared in the Choreography Diagrams and express a decision like Exclusive (XOR) Gateway or Inclusive (OR) Gateway, were also labeled in an adequate manner. The labels indicate the decision that has to be made. Particularly, similar variants of labeling styles like for the Gateways used in Process Diagrams (e.g., the labeling style *Question with Noun and Verb in Past Participle*) are used here too. Parallel (AND) Gateways do not have a label in these Diagrams. However, like in Process diagrams, there is also no need for a label on Parallel (AND) Gateways, because the sequence-flows split and all outgoing paths have to be taken. Hence, there is no need to label this splitting point for expressing a decision that has to be made at this point.

C. Conversation Diagram

The Conversation Diagram was also introduced in Version 2.0 of BPMN. A Conversation Diagram describes the exchange of messages between participants. Once again, these participants are just a continuation of the concept Pool. The graphical notation is a little bit different. Instead of tall rectangles, these participants are modelled with smaller rectangles. The new model element, which is introduced in a Conversation Diagram, is the Conversation and the Conversation Link. The Conversation defines the message exchanges between the Pools. The modeling notation for a Conversation is a hexagon. BPMN distinguishes between normal Conversations, Sub Conversations and Call Conversation. The latter is a reference to a globally defined Conversation or Sub Conversation. Graphically, a Call Conversation is also drawn as a hexagon but the margins are bolder. In order to specify, which participant is connected, to which other participant via a Conversation, the Conversation Links (two parallel lines) connect the participants with Conversations. Figure 11 shows a simple example of a Conversation Diagram.

Since in Conversation Diagrams the participants represent Pools, the modelers mainly use nouns or noun phrases to model these participants. It was also observed that two styles dominate the labels for Conversations. About 42 % of the distinct labels are nouns or noun phrases. In 30 % of the labels, the *Verb Object Style* is used (e.g., "run advertising campaign", "invoke service components").

Additionally, other labeling preferences appear. About 16 % of the labels follow the *Action Noun Style* (e.g., "book reservation"). In the remainder of the cases, the modelers preferred labels, which mainly follow non-canonical label patterns (e.g., "recruitment and training", "delivery/dispatch").

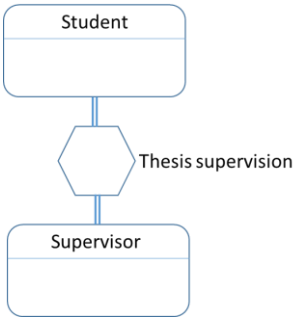


Figure 11. Conversation Diagram

Unfortunately, *Sub Conversations* were rarely used and *Call Conversations* were not used at all in the sample. All that can be said about Sub Conversations is: A dominant labeling style does not exist in this sample.

Since a Conversation represents message exchanges, labeling using a noun, noun phrase or the *Action Noun Style* is more natural. Conversation Links are not labeled. However, this is according to BPMN, which also does not enforce labeling of Conversation Links.

VI. Conclusion and Future Work

This paper described how BPMN model examples are presented on the Web. Particularly, the labels of the model elements of all the four types of BPMN diagrams (Process Diagram, Collaboration Diagram, Choreography Diagram and Conversation Diagram) were examined. In summary, the following can be said about the BPMN model element labels used in Web examples.

For the labels of atomic Activities called Tasks, there is common consensus to follow the recommended *Verb Object Style*, since the majority of the label examples for model elements on the Web follow this style. For non-atomic Activities (i.e., normal Sub processes), two ways of labeling are preferred: Nominalization of a verb (*Action Noun Style*)

and the recommended *Verb Object style*.

In the case of Event types, it turned out that the labeling styles vary depending on the Event type used. Additionally, within the same Event type, variations of labels exist. For some of these labeling strategies, existing literature would even state that these labels have deficiencies. Therefore, it would be good that both providers of such examples and readers of these examples have a more critical look on them. However, it also has to be said that some of the label examples for Events (e.g., Timer Event) are appropriate with respect to the certain type of Event.

If Gateways have a label, then it is quite well understood that a question mark ("?") should close the label, as it is suggested in literature. However, this is the only accordance with literature. Since many of the Gateways do not even have any label. This can be also interpreted as follows. The community, who posts process model examples on the Web, does not yet understand labeling of Gateways as an important feature for a better understanding of the process paths.

It is of common understanding that the Pools (participants) used in Collaboration Diagrams, Choreography Diagrams and Conversation Diagrams as well as the Lanes in Collaboration Diagrams represents systems or role of systems and subsystems, respectively. Therefore, they should be labeled with a nouns or a noun phrase.

In Choreography Diagrams, the labels of the new model element Choreography are consistent with its basis. As already described, the outer sections of a Choreography that represent participating Pools are labeled with nouns or noun phrases. The middle section, which represents the activity triggering or receiving a message, is labeled with the most common labeling styles used in Activities. Namely, it is labeled with the *Verb Object Style* or *Action Noun Style*.

Many of the labels used in the new model element Conversation are either nouns and noun phrases, respectively or they follow the *Verb Object Style*. However, nouns and noun phrases seem to be more natural.

Hence, the new model elements, which appear in Collaboration Diagrams, Choreography Diagrams and Conversation Diagrams, have labels that mainly represent their purpose.

As a future work, it would be interesting to examine Web examples of Process Diagrams, Collaboration Diagrams, Choreography Diagrams and Conversation Diagrams that are modeled with labels in another language than English.

REFERENCES

[1] C. Kop, "Analyzing Model Element Labels of Business Process Model Examples Provided on the Web," The Twelfth International Conference on Information, Process, and Knowledge (eKNOW2020), IARIA ThinkMind, 2020, pp. 28 – 33.

[2] BPMN 2.0 Poster available on the Web: http://www.bpmb.de/images/BPMN2_0_Poster_EN.pdf [retrieved: December 2020].

[3] https://www.ariscommunity.com/users/roland-woldt/2011-01-28-learning-bpmn-2-which-models-are-available-bpmn [retrieved: December 2020].

[4] D. L. Moody, The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notationsin Software Engineering," IEEE Transactions on Software Engineering, vol 35 (6), 2009, pp. 756 – 779.

[5] J. Becker, "Opportunities and Constraints: The Current Struggle with BPMN," Business Process Management Journal, vol. 16 (1), 2010, pp. 181–201.

[6] J. Becker, M. Rosemann, P. Green, and M. Indulska, "Do Ontological Deficiencies in Modeling Grammars Matter?," MIS Quaterly, vol. 35 (1), 2011, pp. 57–79.

[7] B. Silver, BPMN method and style, 2nd edition, Cody-Cassidy Press, 2009.

[8] J. Krogstie, Quality in Business Process Modeling, Springer 2016.

[9] I. Moreno-Montes de Oca, M. Snoeck, and H. A. Reijers, A. Rodriguez-Morffi, "A systematic literature review of studies on business process modeling quality," Journal of Information and Software Technology, vol, 58, 2015, pp. 187-205.

[10] J. Mendling, H. A. Reijers, and W. M. P. van der Aalst, "Seven process modeling guidelines (7PMG)," Journal of Information and Software Technology, vol. 52, 2010, pp. 127–136.

[11] M. Cortes-Cornax, S. Dupuy-Chessa, D. Rieu, and M. Dumas, "Evaluating Choreographies in BPMN 2.0 Using an Extended Quality Framework," International Workshop on Business Process Modeling Notation, Springer, 2009, pp 103-117.

[12] A. Koschmider, M. Ullrich, A. Heine, and A. Oberweis, "Revising the Vocabulary of Business Process Element Labels," International Conference on Advanced Information Systems Engineering (CAiSE 2015), Springer LNCS 9097, 2015, pp. 69–83.

[13] H. Leopold, J. Mendling, and H. A. Reijers, "On the automatic labeling of process models," Proceedings of the 23rd International Conference on Advanced Information Systesm Engineering (CAiSE 2011), Springer LNCS 6741, 2011, pp. 512–520.

[14] K. Figl, J. Mendling, and M. Strembeck, "The Influence of Notational Deficiencies on Process Model Comprehension," Journal of the Association for Information Systems, vol. 14 (6), 2013, pp. 312–338.

[15] A. Koschmider, K. Figl, and A. Schoknecht, "A Comprehensive Overview of Visual Design of Process Model Element Labels," Business Process Management Workshops (BPM 2015, 13th International Workshops), Springer LNBIP 256, 2015, pp. 571–582.

[16] H. Leopold, R. H. Eid-Sabbagh, and J. Mendling, L. Guerreiro Azevodo, F. Araujo Baiao, "Detection of naming convention violations in process models for different languages," Decision Support Systems, vol. 56, 2013, pp. 310-325.

[17] H. Leopold, J. Mendling, and O. Günther, "Learning from Quality Issues of BPMN Models from Industry," IEEE Software, July/August 2016, pp. 26–33.

[18] H. Leopold, H., F. Pittke, and J. Mendling, "Ensuring the Canonicity of process models," Data & Knowledge Engineering, 111, 2017, pp. 22–38.

[19] BPMN 2.0 by Example, Version 1.0 (non-normative), OMG Document Number: dtc/2010-06-02, https://www.omg.org/cgi-bin/doc?dtc/10-06-02.pdf [retrieved: December 2020].

[20] Management Business Process Model and Notation (BPMN), Version 2.0, OMG Document Number formal/2011-01-03, http://www.omg.org/spec/BPMN/2.0 [retrieved: December 2020].

Technology as a Tool to Promote Nontechnical Skills in Surgical Training

Line Lundvoll Warth
Norwegian Centre for E-health Research
Tromsø, Norway
University of Tromsø, The Arctic University of Norway
Tromsø, Norway
email: line.lundvoll.warth@ehealthresearch.no

***Abstract*—Surgeons require strong mentorship as part of their training because many of their skills cannot be readily acquired from textbooks; instead, their competence is a result of excellent hand practice. Access to mentors for education in surgical subspecialties is a challenge in many hospitals. Videoconferencing, which enables real-time communication between mentors and mentees at different geographical locations, can overcome this challenge and make the best knowledge available for surgeons in training. This study examines a practice in Norway in which videoconferencing was used to provide education on a laparoscopic surgical procedure. Specifically, the study explores the characteristics of communication between a mentor and mentee using videoconferencing and how this practice allows for both the learning and feedback of mentorship and nontechnical skills. The empirical material consists of video recordings of an educational trajectory comprising eight patient cases and related focus group meetings. Their communication reveals knowledge gaps and their closure through the establishment of a shared understanding. In this way, videoconferencing supported the learning of technical skills while enabling feedback on nontechnical elements. Both the mentor and mentee were able to reach their full potentials, expanding their own communicative skills and reflecting on their own abilities. Videoconferencing also affected the relationship between the mentor and mentee, who were peers and colleagues rather than participants in a traditional mentee–mentor relationship. Hence, videoconferencing practice is an activity that can expand knowledge and be used to evaluate *both* the mentor and mentee, assessing their nontechnical skills in surgical training.**

Keywords—knowledge sharing; nontechnical skills; surgical training; mentorship; feedback; communication; videoconferencing; qualitative study.

I. INTRODUCTION

Although technical skills in surgery are obviously important, communication in the operating room (OR) plays an important role in patient safety because operations are social situations in which tasks are accomplished through communication between team members. The current study focuses on a practice in Norway during which videoconferencing (VC) was used as a tool for communication in surgical education in a specific laparoscopic hernia procedure. The present paper is an extended version of a paper in which we explored the characteristics of communication between a mentor and mentee using VC and how it affected communication [1]. In the previous paper, we concluded that VC supports the learning of technical skills and enables feedback on nontechnical elements. Both the mentor and mentee had the opportunity to reach their full potentials, expanding their own communicative skills and reflecting on their own abilities. Here, the paper extends that previous work, focusing on the use of VC in relation to nontechnical skills, with a view to the use of VC technology for the learning of technical skills [1] [2] but also as an assessment tool for feedback on the mentor and mentee's nontechnical skills relationship in surgical training.

The life of a surgeon is unique and often challenging. Because surgical training requires skills not readily available from textbooks, surgeons in training require a strong guidance from mentors who can transfer their knowledge to them. A good mentor can be the difference between a surgeon who is skilled and fulfilled and one who is merely competent. The changing surgical environment requires a style of mentorship that is distinct from that in other forms of medicine [3]. This paper argues that VC promotes a style of mentorship in which nontechnical skills can be practised and reflected on, thereby placing greater emphasis on these skills in training. Indeed, the quality of collaboration and teamwork allows for improvements in practice beyond technical skills and performance.

The rest of this paper is organised as follows: Section II explores the field of surgical training, Section III describes the theoretical framework of the study, Section IV describes the methods used, Section V present the results, and Section VI presents the discussion. The article ends with conclusions and acknowledgements.

II. SURGICAL TRAINING

Within surgical teams, communication errors have been studied in terms of communication failures [4], and studies have attempted to explain how surgical procedures are influenced by the quality and efficiency of teamwork. Results have shown that deficiencies in teamwork in the OR contribute significantly to adverse events and patient harm [5] because there is a strong relationship between teamwork failure and technical errors [6]. In other words, a good surgeon is more than just a good 'pair of hands' [7]; he or she must be a good team player, must listen and communicate with colleagues and must empower colleagues to reach their full potential [7]. These qualities are related to

collective and cognitive competence, which are defined as nontechnical skills.

Nontechnical skills are gaining importance in surgery and surgical training [7]. The Royal College of Surgeons of Edinburgh defines nontechnical skills as those skills and behaviours related to situational awareness, decision making, communication, teamwork and leadership [8]. Others have defined nontechnical skills as interpersonal (e.g., communication, teamwork), cognitive (e.g., decision making, situational awareness) and personal resource skills (e.g., coping with stress and fatigue) [9]. Communication and teamwork related to decision making are also important nontechnical skills. All these skills are essential for surgeons to operate safely in the OR, and although they are developed in an informal and tacit manner [8], they need to be explicitly addressed in training.

Surgical training involves the individual work and guidance of an expert mentor. Mentees gain significant skills and experience by participating in simulated environments with virtual simulators and models prior to performing procedures on patients in the OR. Work in the OR involves collaboration; each team member has his or her own tasks to perform. Although each team member's individual technical skills are important, good collaboration is necessary for a good surgical outcome [10][11]. Hence, both mentors and mentees need to develop nontechnical skills in surgical training to promote best practice.

Surgical training is an educational process in which the competence and work of both mentor and mentee serve as parts of a collective activity and communicative process. Both communication and teamwork are important for modern surgical education and practice; indeed, a review of the role of nontechnical skills in surgery showed that the key root cause of surgical errors worldwide is a lack of nontechnical skills [6]. The review also provided evidence that nontechnical skills have an effect on technical performance and suggested that training that is focused on improving nontechnical skills can improve teamwork, performance and safety in the OR, thereby positively contributing to patient outcomes [6]. This indicates that there is a need to focus on the development of nontechnical skills in surgical training.

Because surgery strongly depends on a good pair of hands, surgeons in training are dependent on access to mentors with specialist knowledge. This access to local mentors for surgical subspecialties is a challenge in many hospitals. However, in such cases, VC is a technology that can enable real-time communication between mentors and mentees, even if they are in different geographical locations. Thus, it can help to overcome the issue of a lack of access to local experts.

Research on VC has stressed its educational benefits [12] and has described VC for mentoring as an effective way to develop surgical skills [13]. Recently, however, a review of surgical tele-mentoring reported a limited understanding of VC in surgical practice; the review concluded that little attention has been paid to the educational and nontechnical elements and that focus has instead been placed on piloting the technology [12][14]. Within this field, a special focus on communication and team performance is needed to better understand the factors that influence surgical outcomes [15].

Research on communication in terms of feedback between mentors and mentees reveals that supervisors tend to talk about the trainees' actions and their own frames rather than attempting to understand the trainees' perceptions [16]. Consequently, such comments were only loosely tied to the concrete actions of the trainees. To reach the full potential of feedback, supervisors may benefit from training techniques that would stimulate deeper reflection in trainees [16]. This reflects the need to pay attention to communication and feedback as a two-way knowledge process between mentors and mentees, but communication about both mentors' and mentees' work is not that common.

A wider literature search on communication in the OR concluded that further detailed observational research that provides detailed transcripts and analyses of communication patterns is needed to gain a better understanding of nontechnical skills [17]. Addressing this gap, the current study explores communication and teamwork between a mentor and mentee using VC and the knowledge needed to complete the surgery. The use of VC and the communication between mentor and mentee are followed in real-time surgical training through the educational trajectory of a laparoscopic hernia procedure. Even though it is important to gather information about the outcomes of work in the OR, it is also necessary to gain a detailed understanding of the processes and communication patterns that lead to those outcomes. These are often overlooked in favour of technical skills. Therefore, the current study aims to provide insights into how mentors and mentees organise and accomplish collaborative work using VC in the OR by exploring the characteristics of communication in the relationship between them. It also investigates the feedback in the knowledge sharing between mentor and mentee, focusing on the process of nontechnical skills.

The present study investigates knowledge sharing between a mentor and mentee – specifically, the way in which individual knowledge is shared and constructed to ensure that the mentee applies best practices. It expands upon previous work by exploring VC as an assessment tool for feedback on the activity and nontechnical elements in surgical training.

III. FRAMEWORK

Laparoscopy is a visual technique that uses several small ports in the abdomen, with an instrument inserted through each. The procedure is visual because a small camera is inserted into the patient's abdomen. The images obtained from the camera are transmitted to a monitor in the OR but also enable communication with participants outside the OR. In the cases examined in the current article, the mentee and the surgical team used VC to communicate with a geographically distant mentor. The mentee was experienced in surgery and laparoscopy; before practising this procedure on patients, the mentee underwent the traditional education pathway for a new procedure (i.e., simulations using models and videos of the procedure). The mentor was an international expert in this specific procedure. The surgical

training examined in the current study was organised as three onsite sessions in which the mentor performed, assisted with and observed the procedure and five distant sessions in which the mentor and mentee collaborated.

Communication in the OR was framed using an activity theoretical perspective [18], focusing on the complex interactions between individual subjects and their wider context (i.e., educational activities) [19]. Activity theory is a theoretical framework for analysing and understanding human interactions through the use of tools. The mentor and mentee (subjects) were part of a collaborative educational and communicative process (object) mediated by VC (tool). These elements comprise the individual unit of analysis.

Expanding the unit of analysis of education and learning beyond the individual action [20] includes an additional unit: the activity as the unit of analysis. The sets of conditions (rules) that help determine how and why surgeons act as they do and the distribution of tasks (division of labour) among the community of workers (community) frame the human activity as both individual and collective. Rules and division of labour affect the community; through this, the activity can be analysed. Collaborative activity happens between the activity system of the mentor and mentee, enabling the use of VC in practice. VC is thus a tool that mediates social action (illustrated in Figure 1).

Using activity theory as a framework, educational situations are seen as having a significant historical and cultural context, in which the activity of mentor and mentee is hierarchical in nature and culturally and historically located. The activity is the basic unit of analysis used to understand individual actions in a social context in which the outcome is a new expert and local practice.

IV. METHODS

This is an ethnographic study [21] that explores the use of VC for communication between a mentor and a mentee within an educational process. The study was carried out from 2014 to 2016 in Norway and involved observations, interviews, focus groups and field notes. Five semi-structured interviews, which lasted a total of six hours, took place in 2015 and 2016, and all were transcribed and analysed. For three months in 2014 and 2015, surgical training of the mentee in a specific hernia procedure was observed and videotaped.

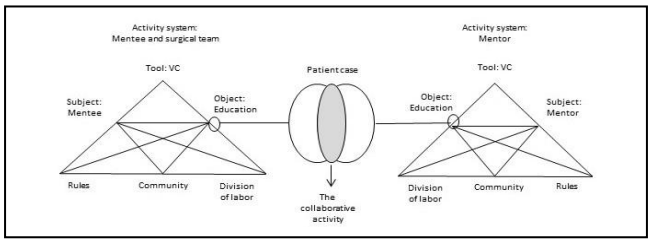


Figure 1. Collaborative activity.

The dataset covers the entire educational trajectory, which includes eight cases and six hours of video observations. The whole dataset was transcribed. All involved participated in two focus group meetings to discuss the procedure. These meetings were also videotaped and transcribed. The mentor was a native English speaker, and the mentee had English as a second language.

The analysis focused on the interactions between the mentor and the mentee, particularly when tensions appeared [21] and knowledge gaps needed to be closed. These interactions shaped the opportunities for expanding verbal decision making and nontechnical skills [22]. The observations in the OR allowed the communication and the team performance to be studied (as opposed to individuals). The eight sessions revealed communication patterns and nontechnical skills (but not individual deficiencies) in a series of operations that utilised VC for educational purposes. The focus group meetings made it possible to study the communication as it arose in the technical performance and reflection.

The study applied for approval from the Regional Committee for Medical and Health Research Ethics, but it was not required for this study. The data-protection officer at the specific hospital approved the study, and all the participants signed an informed consent form.

V. RESULTS

The surgical training examined in the current study was organised into eight sessions. The first three sessions occurred onsite in the OR and involved preparation for the VC, and the next five sessions used VC. After the eighth session, the mentee was considered an expert in this procedure, and the VC sessions ceased [2].

A. Communication using VC

The characteristics of communication using VC are illustrated in Figures 2 and 3.

In Figure 2, we start from the four-minute mark of the seventh session, which was videotaped for about 25 minutes. On the basis of previous sessions, the mentee referred to earlier communication by suggesting a course of action for the day. Specifically, he suggested cauterisation and pulling the sac into the abdominal cavity. He then asked the mentor what he thought about the suggestion (utterance 1). The mentor supported the proposal but had a hunch, based on his own practice with stitches, that simply pulling out the sac would not be adequate (utterance 2).

The mentee referred to the hernia as deep and acknowledged the suggestion to use stitches (utterance 3). The mentor then confirmed that it might be hard to just cauterise (utterance 4). The mentee considered going deep with the instrument (utterance 5), and the mentor elaborated on the depth (utterance 6).

Extract from the seventh session: (A: mentee, B: mentor)

1	A:	I thought maybe today we could try just to cauterise it, if it's possible to – eh – pull the sac out into the abdominal cavity. Or what do you think?
2	B:	Yeah. You can see. You can try. Ehm – it depends. You can try. I always start by turning and, and then if it seems like it's not adequate, then I put a stitch in.
3	A:	It's quite deep, you see …
4	B:	Yeah, it might be hard to do with just cautery.
5	A:	Yeah, I think so to. Because it goes into the …
6	B:	All the way down.
7	A:	Labia majora. Yeah. Okay, I think we will go for …
8	B:	Yeah.
9	A:	I don't think it's even necessary to try. Do you agree?
10	B:	You … but the good thing is, you could do a lot of cautery, you don't have to worry about … Eh ... injuring it.
11	A:	That's good. Okay.

Figure 2. Communication using VC.

After this reflection, the mentee decided to use stiches (utterance 7), a decision that was supported by the mentor (utterance 8). The mentee reconsidered his decision to try to pull the sac into the abdominal cavity and asked the mentor to support this decision (utterance 9). The mentor did support the decision and elaborated on the opportunity to perform cauterisation without injuring the patient (utterance 10). The mentee confirmed that he shared this understanding (utterance 11).

The characteristics of communication in this extract involved skills related to choosing an appropriate course of action and a shared understanding. First, the mentee presented a knowledge gap (i.e., whether to use stiches). This tension between the mentor's knowledge and the mentee's knowledge provided an opportunity to close the knowledge gap, thereby expanding the collective activity of decision making. The mentor supported the suggestion while mentioning the tension between the possible actions (i.e., pulling the sac or using stiches). Drawing on the mentor's experience and knowledge, the mentor and mentee communicated, closing the knowledge gap by establishing a shared understanding. This shared understanding was based on a collective activity in which the participants were able to bridge the gap and perform a successful procedure.

The communication in Figure 3 includes data starting from the 13-minute mark of the eighth session, which was videotaped for about 28 minutes. This extract is a discussion about use of the needle when performing the hernia procedure.

Extract from the eighth session: (A: mentee, B: mentor)

1	B:	You have to go a little bit more medial. So just take the – eh – needle back out a little bit. Then move, and slide in subcuticular (…). Go more medial. Yeah, eh – no, you're too lateral.
2	A:	Still?
3	B:	I can't see the tip of your needle now.
4	A:	You can see it there?
5	B:	Yeah, I think you're … just go … come out of the subcutant a little bit, and just slide the tip of the needle over more medially. Don't be af … Yes, that's better!
6	A:	That's better, yeah.
7	B:	Yeah. Angle it a little … angle it a little more laterally now, so you don't get the epigastrium. Turn it. La … Laterally.
8	A:	It's just – eh – sticking to the peritoneum now.
9	B:	Mhm, just push it, even if it pops out, you can always come back in again.
10	A:	I'm on my way now. There it pops. So, I think maybe just leave the vas.
11	B:	Yeah, I think, I … You're almost there. Just pop, you can pop out.
12	A:	Okay, this was actually one of the – eh – cases that I have learned the most. Because – ehm – the second opening was really tight.
13	B:	Yup!
14	A:	And the thing with the peritoneum vessels and the … it was one of the stickier vasa deferentes I've known.
15	A:	Looks good?
16	B:	That looks great, nice work!

Figure 3. Communication using VC.

The mentor recommends that the mentee move more medially and explains that the mentee needs to take the needle back and out and then slide it in under the skin (subcuticular). Thus, he guides the mentee in the right direction by saying 'go more medial' and 'you're too lateral' (utterance 1). The mentee slides the needle and asks if he is still too lateral (utterance 2). The mentor cannot confirm this because he cannot see the tip of the needle (utterance 3). The mentee moves a bit of the needle under the skin and asks if the mentor can see it now (utterance 4). The mentor, who now sees the needle, recommends that the mentee take the needle back, out of the subcutis, and slide the tip medially (utterance 5). This is a follow-up statement to the mentor's suggestion in utterance 1. The mentee follows the recommendation, and both the mentor and the mentee agree that the method is better than the first method used by the mentee (utterances 5 and 6).

In the new attempt, the mentor guides the mentee by recommending that he 'angle it a little more laterally now' and turn it laterally so that he does not become too close to the epigastrium (utterance 7). The mentee reports that the needle is sticking to the peritoneum (utterance 8). The mentor asks him to push the needle, and by drawing on his experience, he says that the needle can always be brought back if it pops out (utterance 9). The mentee tries to push the needle, and the needle pops out as the mentor said. The mentee suggests leaving the vas (without cauterising) (utterance 10). The mentor supports the mentee about leaving the vas and encourages the mentee by saying 'you are almost there' and recommends that he 'pop out' (utterance 11).

At this moment, the most challenging part of the procedure is over, and the mentee reflects that this case was the one in which he has learned the most from the guidance of the mentor (utterance 12). Since the opening (hernia) was tight, the mentor's knowledge about how to wield the needle was essential for the mentee's method. The mentor confirms that the hernia really was tight (utterance 13) and that the mentee did nice work in this case (utterance 16). The mentee reflects further on why this case was hard: the opening was tight (utterance 12) but also included peritoneal vessels and a sticky vas deferens (utterance 14). The mentee asks if the mentor thinks the result looks good (utterance 15), and the mentor replies 'looks great, nice work' (utterance 16), confirming that the mentee had performed well.

The characteristics of the communication in Figure 3 involve skills related to choosing the right way of using the needle and establishing a shared understanding between the mentor and mentee. First, the mentor offers the mentee knowledge about the method for how to handle the needle (i.e., back and slide in), and the mentee reveals the knowledge gap between his knowledge and the mentee's with regard to the working method. The differences in the methods for using the needle offers an opportunity to close the knowledge gap, thereby expanding the collective activity of decision making. The mentee asks 'still?' and the mentor bridges the knowledge gap by explaining the course of action and establishing a shared understanding. Thus, the knowledge gap is closed through the opportunity to learn a new procedure.

The activity is conducted through the actions of individuals, and by exploring the characteristics of the communication in the relationship between the mentor and mentee, we can obtain insights into how they organise and accomplish collaborative work using VC in the OR. This communication establishes a new work practice.

B. Reviewing the procedure

After each of the eight sessions, the mentor and mentee reviewed the session, as illustrated in Figure 4.

Reviewing a session (A: mentee, B: mentor)

1	B:	I am not sure I like the bend [of your needle].
2	A:	Too much?
3	B:	No, I like it the other way I think.
4	A:	Ah, ok. Yeah, yeah. With all the curve?
5	B:	Yeah, yeah ... Try next time and see if you like it better.

Figure 4. Reviewing the procedure.

When reviewing the session, it came to feedback about the mentee's technique with the needle and how he handled the bend of the needle. The mentor opens up the discussion on the bend of the needle by saying he is not sure whether he likes the bend (utterance 1). The mentee asks if the mentor thinks the bend is too much (utterance 2). The mentor does not refer to the curve of the needle but to an alternative method for handling the needle, 'the other way' (utterance 3). The mentee suggests a method using all the curve length and confirms that he understands that he can handle the needle going with the curve (utterance 4). The mentor verifies that the mentee's suggestion is good and that he can try the other method next time and find out which method he likes best.

Overall, this extract illustrates how the mentor and mentee reflect on their working methods, that is, their technical skills, including methods for using the needle. At the end, the mentor allows the mentee to decide which method he wants to use in his own practice, the mentor's method or the one he performed himself during the procedure.

After the training sessions, focus group meetings was held to review the sessions and allow the mentor and mentee to discuss the content and how VC affected their communication. Figure 5 illustrates how this meeting progressed.

In the excerpt, the mentor asks the mentee about the latter's experience in one of the sessions and how the former could improve as a mentor (utterance 1). The mentee points out the tension between anticipated and 'comfortable' knowledge, referring to the fact that the mentee had watched the training videos of the procedure (utterance 2).

Reviewing two sessions: (A: mentee, B: mentor)

1 B: What was not good? Don't be polite … What could I have done better as a mentor?
2 A: We just assumed that I had seen the video that I knew … You just let me do it, and then you corrected me …
3 B: I didn't give enough instructions (…)? You wish I had given more instructions?
4 A: I don't know if it was necessary, but maybe it would (…) feel more safe, in a way.
5 B: This is a problem that ... Not feeling comfortable as a mentor, knowing not to say too much. When I have a relationship with a resident, I say whatever I want. He is my resident. But when it is a colleague, I am a little bit more shy about being too talkative. Does that make sense? The fact that different relationships exist between me and a trainee, a resident, and another surgeon. I don't want them to be annoyed too much …

Figure 5. Reviewing the procedure.

The mentor asks if the mentee felt that the former had provided too little instruction during the session (utterance 3). Because the session went well, the mentee was not sure whether there was a gap in the knowledge between them but that guidance would have made the mentee feel 'safer' during decision making (utterance 4). The mentor then reflects on the communication between the mentor and mentee, illustrating the tension between the traditional way of locally training mentees (in which the expert mentor holds a more powerful position) and the use of VC as a pre-planned tool for distributed collaborative work, in which the mentor and mentee act as colleagues (utterance 5).

Overall, the extract shows the mentor's and mentee's reflections on their own communicative skills, that is, their nontechnical skills, including how the mentor relates to those around him. By exchanging reflections after the surgical procedure, the mentor was better able to understand his performance as a mentor. This learning activity led to a shared understanding between the activity systems of the mentee and mentor, thereby establishing a new practice for hernia procedures at this hospital.

VI. Discussion

The purpose of the current study was to explore the characteristics of communication between a mentor and mentee using VC in training, and of communication in the feedback about the VC training sessions. By using activity theory as a framework for studying human practices and artefacts in use, the training is understood as a process of development, with both the individual and social levels interlinked. Observing the communication when using VC (Figure 1 and Figure 2) made it possible to identify successful communication and teamwork. This educational process was a collective activity mediated by VC as a cultural tool. Tensions in the work illustrated the limitations of the mentee's individual knowledge, providing opportunities to bridge the knowledge gap between the expert mentor and mentee. Collective decision making led to learning opportunities that allowed the mentee to become an expert in this specific procedure. Thus, communication using VC supported the learning of technical skills.

VC also has the capacity to support collaborative (i.e., nontechnical) skills. The communication examined here refers to previous sessions (a history) and the progress made in expanding the mentee's knowledge. The mentor reflected on his earlier actions and modified his teaching according to the mentee's needs.

The emphasis on decision-making skills in the training allowed the mentee to develop skills related to assessing situations and agreeing on an appropriate course of action within the team. Even though there was a gap in the mentee's knowledge that the mentor had to bridge, the mentor and mentee discussed the options in a balanced way, considering the consequences and benefits of each option and staying flexible while making a shared decision. Afterwards, the mentor explained why he had recommended a specific course of action.

The communication built upon traditional problem-solving in the OR. Laparoscopy is a visual procedure in which a small camera is inserted into the patient's abdomen, and the image is transmitted to a monitor in the OR. In this case, VC was used to show the mentor the same images seen by the mentee. In contrast to traditional training, in which both the mentor and mentee are in the OR, this training occurred using VC. This created tension between the traditional method of local training, in which the mentor and mentee are both at the patient's bedside and are aware of all activity in the OR, and remote guidance, in which the mentor has expert knowledge of the procedure but not complete knowledge of all the activity in the OR.

The problem-solving process is based on the same information, which comes from using the monitor. Consequently, the technical skills are based on the shared knowledge. Nevertheless, there is teamwork in the OR that cannot be experienced by the mentor using VC. Both the mentor and mentee develop awareness of the situation, which includes all the activities in the OR and the pre- and postoperative conditions of the patient. The mentee, who is at the patient's bedside, has the overall picture of the patient. The mentor has expert knowledge and is expected to guide the mentee to deliver high-quality procedures. Thus, because both have great responsibilities, the mentee is more of a colleague than a resident. As the mentor notes in Figure 4, these cases have their own collaborative method that differs from that of traditional mentoring. The mentor acts differently with a colleague than with his own resident, reflecting on what he communicates and trying not to be too talkative and disruptive (i.e., annoying). Using activity theory as a framework, educational situations have a

significant historical and cultural context, such that the activity of the mentor and mentee are hierarchical in nature and culturally and historically located. VC allows the relationship between the mentor and mentee to be more of one between equal colleagues, rather than like the traditional hierarchical mentor–resident relationship. The traditional distribution of tasks and rules are challenged because the VC as a tool mediates social action in a new manner.

When reviewing the procedure, the mentor and mentee discussed both technical skills and the dynamics of the communication patterns (i.e., nontechnical skills). This allowed the mentor to support the mentee while improving his own communication skills through reflection. This activity also supported the mentee in reflecting on his own communication skills. VC was used because the mentee was an experienced surgeon but not in this specific procedure; the competencies of the mentor and mentee were thus unequal in this respect. However, the collaborative activity seemed to affect the historical inequality between the mentor and mentee and redefine the traditional mentee/resident–mentor/expert relationship into one between colleagues. Under the division of labour during surgery, the mentee held the leadership position in the OR, but the mentor was the expert on the procedure. This allowed nontechnical skills, rather than just technical skills, to be developed, subsequently enabling the participants to reflect on how teamwork could be improved.

The use of VC in this setting is not the traditional way of practising training, making it a new tool for this purpose. This may permit more attention to be paid to the problem-solving process and quality than under the traditional way of supervising. Following the trajectory of this training, the team decided to review the technology used to ensure the quality of this supervisory method. This process became more than an evaluation of the technology itself and its capacity for this specific purpose. Reviewing the sessions enabled feedback of the work performed in the VC sessions (Figures 3 and 4). Although it is quite normal to review video films of technical skills during training, it is not that common to include the evaluation of nontechnical activity, that is, the mentor's performance and the communication between the mentor and mentee.

Communication is shaped by organisational culture and historical activities, which play an important role in how work is performed. Communication problems can be attributed to a lack of clarity regarding roles and power relationships [14]. Implementing VC for collaboration in surgical education challenges the traditional surgical training and communication patterns between mentors and mentees. Specifically, the results of the current study illustrate that VC promotes effective reasoning and good communication between mentors and mentees. Communication and teamwork related to decision making are characterised by reflection on the performed work, leading to the development of nontechnical skills and the ability to emphasise nontechnical skills as important in surgical training.

Initially, VC was a tool used to overcome distance. This procedure illustrates how VC has become something more, however, enabling expert knowledge to be shared with mentees who are geographically dispersed. It also illustrates how VC can be used as a tool for feedback on mentorship and collaborative methods. In their study, Entezami et al. [3] called for methods to overcome barriers to effective mentorship, such as a lack of qualified mentors and the lack of an assessment tool to evaluate mentorship in the surgical environment. The present study exemplifies how VC provides a means of assessment for qualified mentors and can educate surgeons, who can then work as mentors for other mentees. In addition, the access to new techniques disperses expert knowledge over geographical distances. Moreover, the current study shows how VC mediates social action, acting as an assessment tool to evaluate mentorship and promote nontechnical skills, encouraging reflection on the communication process. Introducing VC as a tool for communication creates the possibility of offering both traditional and new ways of practising mentorship, enabling the development of an activity for nontechnical skills to become relevant when using VC.

VII. CONCLUSION

In the mentor–mentee relationship studied, contextually embedded interactions occurred between the activity systems of the mentor and mentee. VC allowed knowledge exchange during surgical training, resulting in the mentee becoming an expert in the procedure. The results provide insights into the way in which surgical training and practice are performed, into the communication in training sessions and into the expansion of technical skills.

Because the use of VC as a tool for education in this procedure was new, the surgical team decided to review the technology used to ensure the quality of this method of supervision. This process became more than an evaluation of the technology itself and its capacity for this specific purpose. Reviewing the sessions enabled feedback on the work performed in the VC sessions. The use of VC within surgical training facilitated the development of communication skills because it promoted reflection on both the mentor's and mentee's performance. VC acted as a tool mediating social action, with feedback on the mentee's performance evaluating both the mentor and mentee and assessing the nontechnical skills used in surgical training. The literature has called for an assessment tool to evaluate mentorships in a surgical environment. In this case, VC mediated the evaluation of mentorship and nontechnical skills. Hence, both the mentor and mentee were able to reach their full potential, expanding their own communicative skills and reflecting on their own abilities.

Integrating VC into surgical training within the current training paradigm would allow for both technical and nontechnical elements to be included in the feedback provided to mentees. VC can promote a new style of mentorship in which nontechnical skills can be practised and reflected on while the relevant training is provided. This could be a step towards raising both mentors' and mentees' awareness of nontechnical skills, facilitating changes in the workplace and emphasising collaborative skills (i.e.,

communication and teamwork) in the educational process (and, later, in daily work). In this way, VC could help produce a new generation of surgeons who are competent in all the skills required for knowledge expansion and safe, high-quality patient care.

ACKNOWLEDGEMENTS

Thanks are due to the Northern Regional Health Authority, Norway for funding this project (HST-1181-14) and to all the surgeons who participated in the study.

REFERENCES

[1] L. L. Warth, "Communication between Mentor and Mentee Using Videoconferencing in Surgical Training," IARIA Proc. The Twelfth International Conference on eHealth, Telemedicine, and Social Medicine, 2020, pp. 4–8, ISBN: 978-1-61208-763-4, ISSN 2308-4359. e-ISSN 2308-4359.

[2] L. L. Warth, "Creating learning opportunities by using videoconferencing in surgical education," Stud Health Technol Inform., vol. 262, pp. 15–18, 2019, doi:10.3233/SHTI190005.

[3] P. Entezami, L. E. Franzblau, and K. C. Chung. "Mentorship in surgical training: a systematic review," Hand (NY), vol. 7, no. 1, pp. 30–36, 2012, doi:10.1007/s11552-011-9379-8.

[4] L. Lingard et al., "Communication failures in the operating room: an observational classification of recurrent types and effects," J. Qual. Saf. Health Care, vol. 13, pp. 330–334, 2004.

[5] N. Sevdalis et al., "Quantitative analysis of intraoperative communication in open and laparoscopic surgery," Surg. Endosc., vol. 26, pp. 2931–2938, Oct. 2012.

[6] L. Hull et al., "The impact of nontechnical skills on technical performance in surgery: a systematic review," J. Am. Coll. Surg., vol. 214, pp. 214–230, 2012, doi:10.1016/j.jamcollsurg. 2011.10.016.

[7] R. A. Agha, A. J. Fowler, and N. Sevdaliscet, "The role of non-technical skills in surgery," Ann. Med. Surg., vol. 4, pp. 422–427, Published online 2015 Oct 9. doi: 10.1016/j.amsu. 2015.10.006.

[8] The Royal College of Surgeons of Edinburgh. *NOTSS: Non-technical Skills for Surgeons*. [Online]. Available from: http://www.abdn.ac.uk/iprc/notss/ 2020.01.21.

[9] R. Flin, P. O'Connor, and M. Crichton, Safety at the Sharp End: A Guide to Nontechnical Skills. Aldershot: Ashgate, 2008.

[10] B. S. Nedrebo et al., "Survival effect of implementing national treatment strategies for curatively resected colonic and rectal cancer," Br. J. Surg., vol. 98, pp. 716–723, 2011.

[11] N. J. Birkmeyer et al., "Safety culture and complications after bariatric surgery," Ann. Surg., vol. 257, pp. 260–265, 2013.

[12] K. M. Augestad et al., "Surgical telementoring in knowledge translation—clinical outcomes and educational benefits: a comprehensive review," Surg. Innov., vol. 20, no. 3, pp. 273–281, 2013.

[13] P. Lamba, "Teleconferencing in medical education: a useful tool," Australas. Med. J., vol. 4, pp. 442–447, 2011.

[14] L. Panait et al., "Telementoring versus on-site mentoring in virtual reality-based surgical training," Surg. Endosc., vol. 20, pp. 113–118, 2006.

[15] K. M. Augestad et al., "Educational implications for surgical telementoring: a current review with recommendations for future practice, policy, and research," Surg. Endosc., vol. 31, pp. 3836–3846, 2017.

[16] L. Spanager, P. Dieckmann, R. Beier-Holgersen, J. Rosenberg, and D. Oestergaard, "Comprehensive feedback on trainee surgeons' non-technical skills," International Journal of Medical Education, vol. 6, pp. 4–11, 2015. ISSN: 2042-6372, doi:10.5116/ijme.54b4.2196.

[17] C. Vincent, K. Moorthy, S. Sarker, A. Chang, and A. W. Darzi, "Systems approaches to surgical quality and safety: from concept to measurement," Ann. Surg., vol. 239, pp. 475–482, 2004.

[18] Y. Engeström, "Expansive learning at work: towards an activity theory reconceptualization," J. Educ. Work, vol. 14, pp. 133–156, 2001.

[19] S. M. Weldon, T. Korkiakangas, J. Bezemer, and R. Kneebone, "Communication in the operating theatre," Br. J. Surg., vol. 100, pp. 1677–1688, Dec. 2013, doi:10.1002/bjs.9332.

[20] Y. Engeström and H. Kerosuo, "From workplace learning to inter-organizational learning and back: the contribution of activity theory. Guest editorial," J. Workplace Learn., vol. 19, pp. 336–342, 2007.

[21] D. Silverman, Interpreting Qualitative Data: Methods for Analysing Talk, Text and Interaction. London: Sage Publications, 2001.

[22] P. Linell, Approaching Dialogue: Talk, Interaction and Contexts in Dialogical Perspective. Amsterdam: John Benjamins, 1998.