IARIA

SOFTENG 2021

The Seventh International Conference on Advances and Trends in Software Engineering

ISBN: 978-1-61208-843-3

April 18 - 22, 2021

SOFTENG 2021 Editors

Tugkan Tuglular, Izmir Institute of Technology, Turkey

Manuela Popescu, IARIA, EU/USA

SOFTENG 2021

Forward

The Seventh International Conference on Advances and Trends in Software Engineering (SOFTENG 2021) continued a series of events focusing on the challenging aspects for software development and deployment, across the whole life-cycle.

Software engineering exhibits challenging dimensions in the light of new applications, devices and services. Mobility, user-centric development, smart-devices, e-services, ambient environments, e-health and wearable/implantable devices pose specific challenges for specifying software requirements and developing reliable and safe software. Specific software interfaces, agile organization and software dependability require particular approaches for software security, maintainability, and sustainability.

We take here the opportunity to warmly thank all the members of the SOFTENG 2021 technical program committee, as well as all the reviewers. The creation of such a high quality conference program would not have been possible without their involvement. We also kindly thank all the authors who dedicated much of their time and effort to contribute to SOFTENG 2021. We truly believe that, thanks to all these efforts, the final conference program consisted of top quality contributions. We also thank the members of the SOFTENG 2021 organizing committee for their help in handling the logistics of this event.

SOFTENG 2021 Chairs

SOFTENG 2021 Steering Committee
Paolo Maresca, VERISIGN, Switzerland

SOFTENG 2021 Advisory Committee
Mira Kajko-Mattsson, Royal Institute of Technology, Sweden
Yoshihisa Udagawa, Tokyo Polytechnic University, Japan
Ulrike Hammerschall, University of Applied Sciences Munich, Germany

SOFTENG 2021 Industry/Research Advisory Committee
Philipp Helle, Airbus Group Innovations - Hamburg, Germany
Tomas Schweigert, SQS Software Quality Systems AG, Germany
Michael Perscheid, Innovation Center Network, SAP, Germany
Doo-Hwan Bae, Software Process Improvement Center - KAIST, South Korea

SOFTENG 2021 Publicity Chairs
Lorena Parra, Universitat Politecnica de Valencia, Spain
Jose Luis García, Universitat Politecnica de Valencia, Spain

SOFTENG 2021

Committee

SOFTENG 2021 Steering Committee

Paolo Maresca, VERISIGN, Switzerland

SOFTENG 2021 Advisory Committee

Mira Kajko-Mattsson, Royal Institute of Technology, Sweden
Yoshihisa Udagawa, Tokyo Polytechnic University, Japan
Ulrike Hammerschall, University of Applied Sciences Munich, Germany

SOFTENG 2021 Industry/Research Advisory Committee

Philipp Helle, Airbus Group Innovations - Hamburg, Germany
Tomas Schweigert, SQS Software Quality Systems AG, Germany
Michael Perscheid, Innovation Center Network, SAP, Germany
Doo-Hwan Bae, Software Process Improvement Center - KAIST, South Korea

SOFTENG 2021 Publicity Chairs

Lorena Parra, Universitat Politecnica de Valencia, Spain
Jose Luis García, Universitat Politecnica de Valencia, Spain

SOFTENG 2021 Technical Program Committee

Khelil Abdelmajid, Landshut University of Applied Sciences, Germany
Mo Adda, University of Portsmouth, UK
Bestoun S. Ahmed, Karlstad University, Sweden
Issam Al-Azzoni, Al Ain University of Science and Technology, UAE
Vahid Alizadeh, College of Computing & Digital Media - DePaul University, USA
Washington Almeida, Cesar School | Center of Advanced Studies and Systems of Recife, Brazil
Mohamed Basel Almourad, College of Technological Innovation - Zayed University, Dubai, UAE
Hussein Almulla, University of South Carolina, USA / University of Anbar, Irak
Vu Nguyen Huynh Anh, Université Catholique de Louvain, Belgium
Darlan Arruda, University of Western Ontario, Canada
Jocelyn Aubert, Luxembourg Institute of Science and Technology (LIST), Luxembourg
Lerina Aversano, University of Sannio, Italy
Ali Babar, University of Adelaide, Australia
Doo-Hwan Bae, Software Process Improvement Center - KAIST, South Korea
Musard Balliu, KTH Royal Institute of Technology, Sweden
Imen Ben Mansour, University of Manouba, Tunisia
Marciele Berger, University of Minho, Portugal

Marcello M. Bersani, Politecnico di Milano, Italy
Anna Bobkowska, Gdansk University of Technology, Poland
Luigi Buglione, Engineering Ingegneria Informatica SpA, Italy
Azahara Camacho, University of Cádiz, Spain
Qinglei Cao, University of Tennessee, Knoxville, USA
Pablo Cerro Cañizares, Universidad Complutense de Madrid, Spain
Sang Kil Cha, KAIST, Korea
Allaoua Chaoui, University Constantine 2 - Abdelhamid Mehri, Algeria
Andrea D'Ambrogio, University of Rome Tor Vergata, Italy
Luciano de Aguiar Monteiro, Institute of Higher Education iCEV – Teresina-Piauí, Brazil
Amleto Di Salle, University of L'Aquila, Italy
Sigrid Eldh, Ericsson AB, Sweden
Gencer Erdogan, SINTEF Digital, Norway
Fernando Escobar, PMI-DF Brasilia, Brazil
Faten Fakhfakh, National School of Engineering of Sfax, Tunisia
Barbara Gallina, Mälardalen University, Sweden
Atef Gharbi, National Institute of Applied. Sciences and Technology, Tunisia
Pablo Gordillo, Universidad Complutense de Madrid, Spain
Jiaping Gui, NEC Laboratories America Inc., USA
Adriana Guran, Babes-Bolyai University, Cluj-Napoca, Romania
Ulrike Hammerschall, University of Applied Sciences Munich, Germany
Xue Han, University of Southern Indiana, Evansville, USA
Noriko Hanakawa, Hannan University, Japan
Qiang He, Swinburne University of Technology, Australia
Philipp Helle, Airbus Group Innovations - Hamburg, Germany
Samedi Heng, Université de Liège, Belgium
Jang Eui Hong, Chungbuk National University, South Korea
Fu-Hau Hsu, National Central University, Taiwan
LiGuo Huang, Southern Methodist University, USA
Rui Humberto Pereira, ISCAP/IPP, Portugal
Carlos Hurtado Sánchez, Tecnológico Nacional de México - campus Tijuana, Mexico
Shinji Inoue, Kansai University, Osaka, Japan
Anca Daniela Ionita, University Politehnica of Bucharest, Romania
Jiajun Jiang, Tianjin University, China
Mira Kajko-Mattsson, Royal Institute of Technology, Sweden
Atsushi Kanai, Hosei University, Japan
Carlos Kavka, ESTECO SpA, Trieste, Italy
Wiem Khlif, Mir@cl Laboratory | University of Sfax, Tunisia
Alexander Knapp, Universität Augsburg, Germany
Johann Krautlager, Airbus Defence and Space GmbH, Germany
Dieter Landes, University of Applied Sciences Coburg, Germany
Bruno Lima, INESC TEC | FEUP, Porto, Portugal
Seyong Lee, Oak Ridge National Laboratory, USA
Maurizio Leotta, University of Genova, Italy
Damian M. Lyons, Fordham University, USA
Qinghua Lu, CSIRO, Australia
Yingjun Lyu, University of Southern California, USA
Jianbing Ma, Chengdu University of Information Technology, China

Eda Marchetti, ISTI-CNR, Pisa, Italy
Paolo Maresca, VERISIGN, Switzerland
Imen Marsit,University of Sousse, Tunisia
Mahshid Helali Moghadam, RISE Research Institutes of Sweden, Sweden
Edgardo Montes de Oca, Montimage, Paris, France
Fernando Moreira, Universidade Portucalense, Portugal
Ines Mouakher, University of Tunis El Manar, Tunisia
Mohammad Reza Nami, Islamic Azad University-Qazvin, Iran
Krishna Narasimhan, Itemis AG, Stuttgart, Germany
Risto Nevalainen, FiSMA (Finnish software measurement association), Finland
Virginia Niculescu, Babes-Bolyai University, Cluj-Napoca, Romania
Stoicuta Olimpiu, University of Petrosani, Romania
Rafael Oliveira, UTFPR - The Federal University of Technology - Paraná, Brazil
Luca Pascarella, Delft University ofTechnology, Netherlands
João Pascoal Faria, University of Porto, Portugal
Antonio Pecchia, Università degli Studi di Napoli Federico II, Italy
Fabiano Pecorelli, University of Salerno, Italy
Michael Perscheid, SAP Technology & Innovation, Germany
Dessislava Petrova-Antonova, Sofia University, Bulgaria
Stefano Quer, Politecnico di Torino, Italy
Fumin Qi, National Supercomputing Center in Shenzhen (Shenzhen Cloud Computing Center), China
Zhengrui Qin, Northwest Missouri State University, USA
Aamir Raihan, University of British Columbia, Canada
Oliviero Riganelli, University of Milan-Bicocca, Italy
Michele Risi, University of Salerno, Italy
Simona Riurean, University of Petrosani, Romania
António Miguel Rosado da Cruz, Higher School Technology and Management - Polytechnic Institute of Viana do Castelo, Portugal
Gunter Saake, Otto-von-Guericke-Universitaet, Magdeburg, Germany
Sébastien Salva, University Clermont Auvergne, France
Hiroyuki Sato, University of Tokyo, Japan
Daniel Schnetzer Fava, University of Oslo, Norway
Tomas Schweigert, SQS Software Quality Systems AG, Germany
Ali Sedaghatbaf, RISE Research Institute of Sweden, Sweden
Alberto Sillitti, Innopolis University, Russia
Rocky Slavin, University of Texas at San Antonio, USA
Cristovão Sousa, Polytechnic Institute of Porto / INESC TEC, Portugal
Bernard Stepien, University of Ottawa, Canada
Ahmed Tamrawi, EnSoft Corp., USA
Yoshihisa Udagawa, Tokyo University of Information Sciences, Japan
Miroslav Velev, Aries Design Automation, USA
Colin Venters, University of Huddersfield, UK
Roberto Verdecchia, Gran Sasso Science Institute (GSSI), Italy /Vrije Universiteit Amsterdam (VU), Netherlands
Flavien Vernier, Université Savoie Mont Blanc, France
László Vidács, University of Szeged, Hungary
António Vieira, University of Minho, Portugal
Shaohua Wang, New Jersey Institute of Technology, USA

Ralf Wimmer, Concept Engineering GmbH / Albert-Ludwigs-Universität Freiburg, Freiburg im Breisgau, Germany
Xiaofei Xie, Nanyang Technological University, Singapore
Cemal Yilmaz, Sabanci University, Istanbul, Turkey
Levent Yilmaz, Auburn University, USA
Peter Zimmerer, Siemens AG, Germany
Alejandro Zunino, ISISTAN, UNICEN & CONICET, Argentina

Copyright Information

Table of Contents

A Historical and Statistical Study of the Software Vulnerability Landscape

Assane Gueye
Carnegie Mellon
University Africa
Kigali, Rwanda
assaneg@andrew.cmu.edu

Peter Mell
National Institute
of Standards and Technology
Gaithersburg MD, USA
peter.mell@nist.gov

***Abstract*—Understanding the landscape of software vulnerabilities is key for developing effective security solutions. Fortunately, the evaluation of vulnerability databases that use a framework for communicating vulnerability attributes and their severity scores, such as the Common Vulnerability Scoring System (CVSS), can help shed light on the nature of publicly published vulnerabilities. In this paper, we characterize the software vulnerability landscape by performing a historical and statistical analysis of CVSS vulnerability metrics over the period of 2005 to 2019 through using data from the National Vulnerability Database. We conduct three studies analyzing the following: the distribution of CVSS scores (both empirical and theoretical), the distribution of CVSS metric values and how vulnerability characteristics change over time, and the relative rankings of the most frequent metric value over time. Our resulting analysis shows that the vulnerability threat landscape has been dominated by only a few vulnerability types and has changed little during the time period of the study. The overwhelming majority of vulnerabilities are exploitable over the network. The complexity to successfully exploit these vulnerabilities is dominantly low; very little authentication to the target victim is necessary for a successful attack. And most of the flaws require very limited interaction with users. However, on the positive side, the damage of these vulnerabilities is mostly confined within the security scope of the impacted components. A discussion of lessons that could be learned from this analysis is presented.**

***Index Terms*—Vulnerabilities, Statistics**

I. Introduction

Understanding the landscape of software vulnerabilities is a key step for developing effective security solutions. It is difficult to counter a threat that is not well understood. Fortunately, there exist vulnerability databases that can be analyzed to help shed light on the nature of publicly published software vulnerabilities. The National Vulnerability Database (NVD) [1] is one such repository. NVD catalogs publicly disclosed vulnerabilities and provides an analysis of their attributes and severity scores using the Common Vulnerability Scoring System (CVSS) [2]. CVSS is used extensively by security tools and databases and is maintained by the international Forum of Incident Response and Security Teams (FIRST) [3].

The CVSS provides a framework for describing vulnerability attributes and then scoring them as to their projected severity. The attributes are metric values that are the input to a CVSS equation that generates the score. It is the vulnerability attribute descriptions (the metric values) that are of primary interest to our work, although we also look at the raw scores. The use of CVSS by vulnerability databases provides a suite of low level metrics, encapsulated in a vector, describing the characteristics of each vulnerability. CVSS was initially released in 2005 [4], was completely revamped with version 2 (v2) in 2007 [5], and was updated with new and modified metrics in 2015 with the release of version 3 (v3) [6]. Note that a minor update version 3.1 was released in 2019 [7], but the changes therein do not affect our work. The software flaw vulnerability landscape was thoroughly analyzed in the scientific literature using v2 when it was first released [4], [8]–[13], but little work has been done since to evaluate changes to that landscape over time. Also in our literature survey, we did not find a single study that uses the updated and significantly modified v3 to understand the software vulnerability landscape.

In this paper, we use the CVSS v2 and v3 data provided by the NVD to undertake a historical and statistical analysis of the software vulnerabilities landscape over the period 2005 to 2019. More precisely, we conduct three studies analyzing the following:

- score distributions,
- metric value distributions,
- and relative rankings of the most frequent metric values.

For our first study, we analyze and compare the distributions of CVSS v2 and v3 scores as generated from the NVD data. We then compare the empirical distributions against the theoretical score distributions, assuming that all CVSS vectors are equally likely (which is not the case, but it is illustrative to evaluate the differences).

For our second study, we compute the distributions of the CVSS metric values (i.e., vulnerability characteristics) for each year. We then analyze the differences from 2005 to 2019 to determine if and how vulnerability characteristics change over time.

For our third study, we identify the most frequent metric values and analyze their relative rankings from 2015 to 2019. For each year and for both CVSS versions, we compute the values of the top 10 observed vulnerability metrics as well as their frequencies. We then generate parallel coordinates plots showing the values and frequencies of each metric for each year.

Our analysis shows that the software vulnerability landscape has been dominated by only a few vulnerability types and has changed very little from 2005 to 2019. For example, the overwhelming majority of vulnerabilities are exploitable over the network (i.e., remotely). The complexity to successfully exploit these vulnerabilities is dominantly low while attackers are generally not required to have any level of prior access to their targets (i.e., having successfully authenticated) in order to launch an attack. And most of the flaws require very limited interaction with users. On the positive side, the damage of these vulnerabilities is mostly confined within the security scope of the impacted components. Few vulnerabilities obtain greater privileges than are available to the exploited vulnerable component.

Our findings are consistent with previous studies [8] (mainly based on CVSS version 2). This indicates that the same vulnerabilities are still being found in our software, suggesting that the community has not been doing a great job correcting the most common vulnerabilities.

The remainder of this paper is organized as follows. Section II presents the CVSS data sets that constitute the basis of our study. Section III gives the details of our analysis and our discussion. Section IV provides a summary of related work and Section V concludes.

II. The CVSS Datasets

CVSS consists of three metric groups: base, temporal, and environmental. The base group represents the intrinsic qualities of a vulnerability that are constant over time and across user environments, the temporal group reflects the characteristics of a vulnerability that change over time, and the environmental group represents the characteristics of a vulnerability that are unique to a user's environment [6]. In this work, we evaluate only the base metrics as no extensive database of temporal scores exists and the environment metrics are designed for an organization to customize base and temporal scores to their particular environment.

Tables I and II show the base score metrics and possible values for v2 and v3, respectively. The CVSS base score takes into account the exploitability (how easy it is to use the vulnerability in an attack) and impact (how much damage the vulnerability can cause to an affected component) of a vulnerability apart from any specific environment.

The exploitability score is determined by the following:

- attack vector (v2 & v3): 'the context by which vulnerability exploitation is possible',
- attack complexity (v2 & v3): 'the conditions beyond the attacker's control that must exist in order to exploit the vulnerability',
- authentication (v2): 'number of times an attacker must authenticate to a target in order to exploit a vulnerability',
- privileges required (v3): 'the level of privileges an attacker must possess before successfully exploiting the vulnerability', and
- user interaction (v3): a human victim must participate for the vulnerability to be exploited.

TABLE I
CVSS v2 METRICS

CVSS v2 Metrics	Metric Values
Access Vector (AV)	Network (N), Adjacent (A), Local (L)
Attack Complexity (AC)	Low (L), Medium (M), High (H)
Authentication (Au)	Multiple (M), Single (S), None (N)
Confidentiality (C)	Complete (C), Partial (P), None (N)
Integrity (I)	Complete (C), Partial (P), None (N)
Availability (A)	Complete (C), Partial (P), None (N)

TABLE II
CVSS v3 METRICS

CVSS v3 Metrics	Metric Values
Attack Vector (AV)	Network (N), Adjacent (A), Local (L), Physical (P)
Attack Complexity (AC)	Low (L), High (H)
Privileges Required (PR)	None (N), Low (L), High (H)
User Interaction (UI)	None (N), Required (R)
Scope (S)	Unchanged (U), Changed (C)
Confidentiality (C)	High (H), Low (L), None (N)
Integrity (I)	High (H), Low (L), None (N)
Availability (A)	High (H), Low (L), None (N)

The impact score (v2 & v3) is determined by measuring the impact to the confidentiality, integrity, and availability of the affected system. Also included (v3) is a scope metric that 'captures whether a vulnerability in one vulnerable component impacts resources in components beyond its security scope'.

A particular assignment of metric values is then used as input to the CVSS base score equations to generate scores representing the inherent severity of a vulnerability in general, apart from any particular environment. The raw score in the range from 0 to 10 is then often translated into a 'qualitative severity rating scale' (None: 0.0, Low: 0.1 to 3.9, Medium: 4.0 to 6.9, High: 7.0 to 8.9, and Critical: 9.0 to 10.0) [6].

Vulnerability analysts apply the metrics to vulnerabilities to generate CVSS vector strings. The vectors describe the metric values, but not the CVSS scores, for a particular vulnerability using a simplified notation.

The NVD is the 'U.S. government repository of standards based vulnerability management data' [1]. It provides CVSS vectors and base scores for all vulnerabilities listed in the Common Vulnerabilities and Exposures (CVE) [14] [15] catalog of publicly disclosed software flaws. We use NVD to evaluate both CVSS v2 and v3 vectors and scores. The v2 data covers all CVE vulnerabilities published between 2005 and 2019. The v3 data ranges from 2015 to 2019 (only limited v3 data is available prior to 2015). These coverage dates result in the inclusion in our study of 118 173 v2 vectors and scores and 55 441 v3 vectors and scores.

III. Data Analysis

We analyze the NVD CVSS data in order to better understand the software vulnerability landscape. We investigate both the current nature of the threat posed by the existence and public disclosure of these vulnerabilities as well as how this threat has changed over time. To achieve this, we conduct the three studies described previously where we analyze the following: score distributions, metric value distributions, and relative rankings of the most frequent metric values.

A. Score Distributions

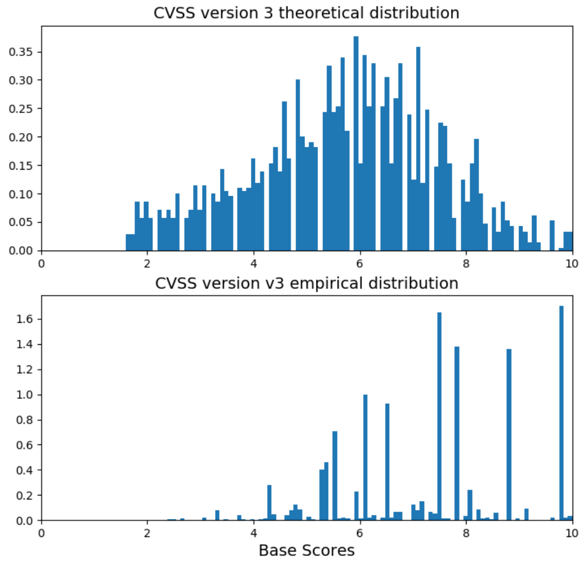


Fig. 1. Theoretical vs Empirical Score Distributions for CVSS version 3. The y-axis shows the numerical values of the base scores of vulnerabilities. The top figure is obtained by considering all possible assignments of metric values, while the bottom figure corresponds to scores of actual vulnerabilities discovered in software.

The top graph of Figure 1 shows the theoretical distribution of the v3 scores (v2 scores are similar and not shown in the paper due to space limitation. They can be found in the appendix of [16]). These plots show what is expected if all CVSS vectors (i.e., vulnerability types) are equally likely to occur. Note how the theoretical distribution was designed, by the FIRST CVSS committee, to spread CVSS scores throughout the range with a somewhat normal distribution with the most probable scores occurring in the middle of the distribution (a little biased to the right). That said, it is interesting in that for both v2 and v3 some scores are not possible even though they lie within the valid range of score values.

The empirical distribution is shown in the bottom of Figure 1 for v3. The empirical data indicates a predominance of certain vectors (groupings of vulnerability characteristics) in the real world. Thus, only a few vulnerability feature sets describe the majority of publicly disclosed vulnerabilities. This leads to the frequent use of just a very small number of scores. A similar observation was made in a previous study of the v2 scoring system [8].

The results observed with v3, which uses data from 2015 to 2019 (since v3 vectors are not generally available prior to 2015) are similar to those with v2, which uses data from 2005 to 2019. Hence, the long-term obtained with CVSS v2 data is confirmed by the shorter-term data of CVSS v3.

B. Metric Value Distributions

To investigate more carefully (in order to identify) possible differences per year and trends over time, we focus on the distributions of each set of metric values per year over the time period of study. Figure 2 provides the histograms for v3 from 2015 to 2019. We have also plotted the histograms for v2 [16], which cover from 2005 to 2019. The inclusion of v2 in the study allows for a comparison over 15 years as opposed to being limited to just 5 years with v3, due to its more recent development.

The histograms for individual metric values for v3 appear almost the same year to year for the 5 years of study. This is the same in v2 over the longer time period of 15 years with some small exceptions: in 2014, the attack vector (AV) value of adjacent had some significance. According to the NVD team [17], this was a one time anomaly due to more than 800 CVEs all being announced simultaneously by an organization doing analyses on phone apps. The Attack Complexity (AC) value ‘Medium’ increased some from 2007 onward, but then was steady, the Authentication (Au) value ‘Single’ increased slightly over the years, and the Confidentiality (C), Integrity (I), and Availability (A) metric proportions between ‘None’, ‘Partial’, and ‘Complete’ varied slightly from year to year while generally maintaining themselves about the same.

Overall though, the software vulnerability landscape for publicly disclosed vulnerabilities has been almost static during the period of study. This said, doing comparisons between the v2 and v3 histograms, we see some differences, but this is due to differences in the approaches of the two versions of CVSS. These differences are primarily seen in regards to the metrics C, I, and A, which we will discuss shortly.

Consider the AV metric which reflects the context by which the vulnerability can possibly be exploited: Network (N), Adjacent (A), Local (L), or Physical (P). Both data sets show a high peak at N, a low peak at L and almost nothing at A and P. This indicates that the overwhelming majority of publicly disclosed software vulnerabilities are exploitable over the network (i.e., remotely), and it has been that way consistently through the period of study.

The AC metric describes the conditions beyond the attacker’s control that must exist in order to exploit the vulnerability. When it is low (AC:L), the attacker can expect repeatable easy successes, while when it is high (AC:H) the attack is less likely to be successful. The data shows that the AC metric is largely dominated by the values of AC:L for v3 and AC:L and AC medium (AC:M) for v2. This indicates that the set of publicly disclosed vulnerabilities have been predominantly easy to exploit.

This “easiness” to exploit vulnerabilities is confirmed by the other metrics for each CVSS version. For v3, the Privileges Required (PR) metric describes the level of privileges an attacker must possess before successfully exploiting a vulnerability. The User Interaction (UI) metric captures the requirements for a human user (other than the attacker) to participate in the successful compromising of the vulnerable

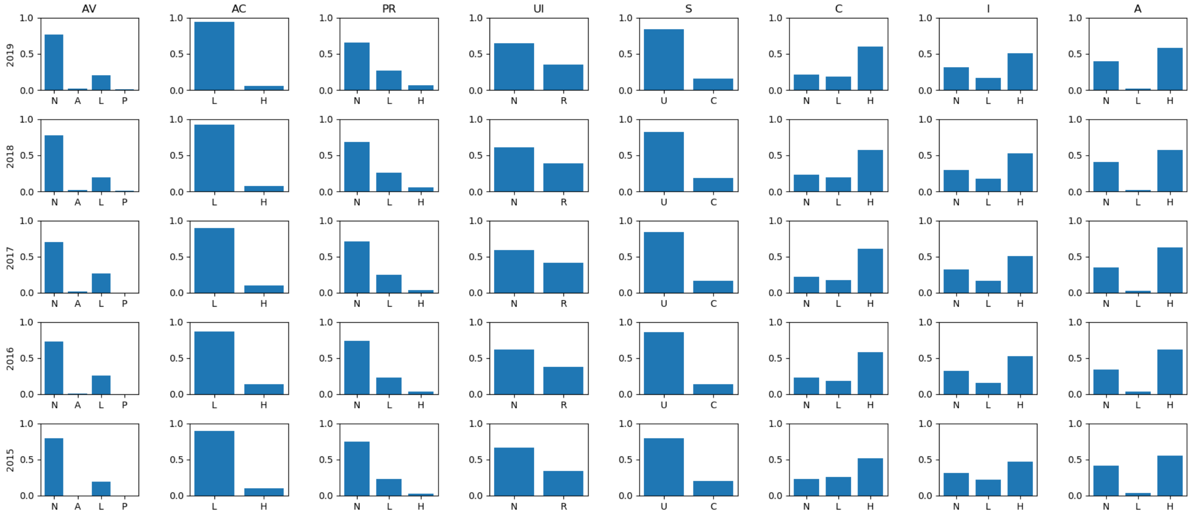


Fig. 2. CVSS v3 metrics' values distributions over the years

components. The data shows that in most of the cases, no privilege is required and very little user interaction is needed for a successful attack.

Similarly, with v2, the Au metric measures the number of times an attacker must legitimately authenticate to a target in order to be in a position to exploit a vulnerability. The data shows that almost always, there is no authentication required prior to exploiting a vulnerability. Sometimes a single authentication is required, but almost never is there a vulnerability that requires multiple authentications in order to be successfully exploited.

CVSS v3 introduced a new Scope (S) metric, which captures the spill-over effect: how much a vulnerability in one vulnerable component impacts resources in components outside of its security scope. When the scope is unchanged (S:U), there is no spill-over, while when the scope is changed (S:C) the vulnerability will very likely affect other components. The data shows that the scope metric has predominantly been S:U.

The last three metrics C, I, and A are common to both CVSS versions. They capture the extent to which a successful exploitation of a vulnerability will affect these three principles of security on the effected component. With respect to these metrics, the v3 data shows that the impact on C, I, and A has predominantly been high (C:H, I:H, and A:H) with very similar distributions for all the years. The v2 data also shows a similar stationary behavior in the distributions. However, the difference in the fraction of high for v3 and complete for v2 is notable. One might expect the high values in CVSS v3 to be equivalent to the complete values for v2. However, this is not the case as they are defined differently. According to the NVD team [17] "the CVSS scoring systems are fundamentally different regarding qualifications for CIA Partial/Complete and Low/High. This is a common misconception due to the nuances of the scoring systems. In addition to this, the NVD takes the approach of representing the worst-case scenario when information is lacking. This typically results in default values of HIGH being attributed to a CVE unless data is available that identifies a limitation to the impact or meets qualifying text for the specification."

C. Relative Rankings of the Most Frequent Metric Values

We now focus on the most prominent individual values of the metrics, evaluating the rankings of the top 10 metric values observed each year and providing a comparison between the years. Figure 3 shows the rankings for v3 (the same plots for v2 can be found here [16]). The y-axes show the top 10 most prevalent metric values, ordered from the least frequent to most frequent. Thus, the set of metric values included in the y-axis is significant (only the top ten are shown). The x-axes show the years. Each (x,y) point indicates that in year x the metric value at y has a rank indicated by the number in the circle. The size of the circle is proportional to the number of times that metric value appeared in a score in that year. For example in Figure 3, in 2017, the metric value AV-N was the fourth most frequent metric value within the set of all v3 vectors. However, in 2018 and 2019 this metric value became the third most frequent. Notice that in general, a value might appear in the top 10 of one year while not appearing in another year. Whenever that happens, we rank that particular value 11 for all the years in which it did not appear.

For v3 (see Figure 3), we observed that the same top 10 values appeared from 2016 to 2019. Furthermore, only one of those values is missing in the 2015 top 10. In addition, these values were ranked almost the same over the years. The top 2 are constant and in the same order over the time period 2015 to 2019. The top 4 and the bottom 4 (including the 11th appended value) are also constant with minor changes in the order they appear over the years. The v2 data shows similar results (see [16]). This is another illustration of the stationary threat landscape observed earlier. It also corroborates the observations in Figure 1, that the landscape has been dominated by just a few vulnerability types.

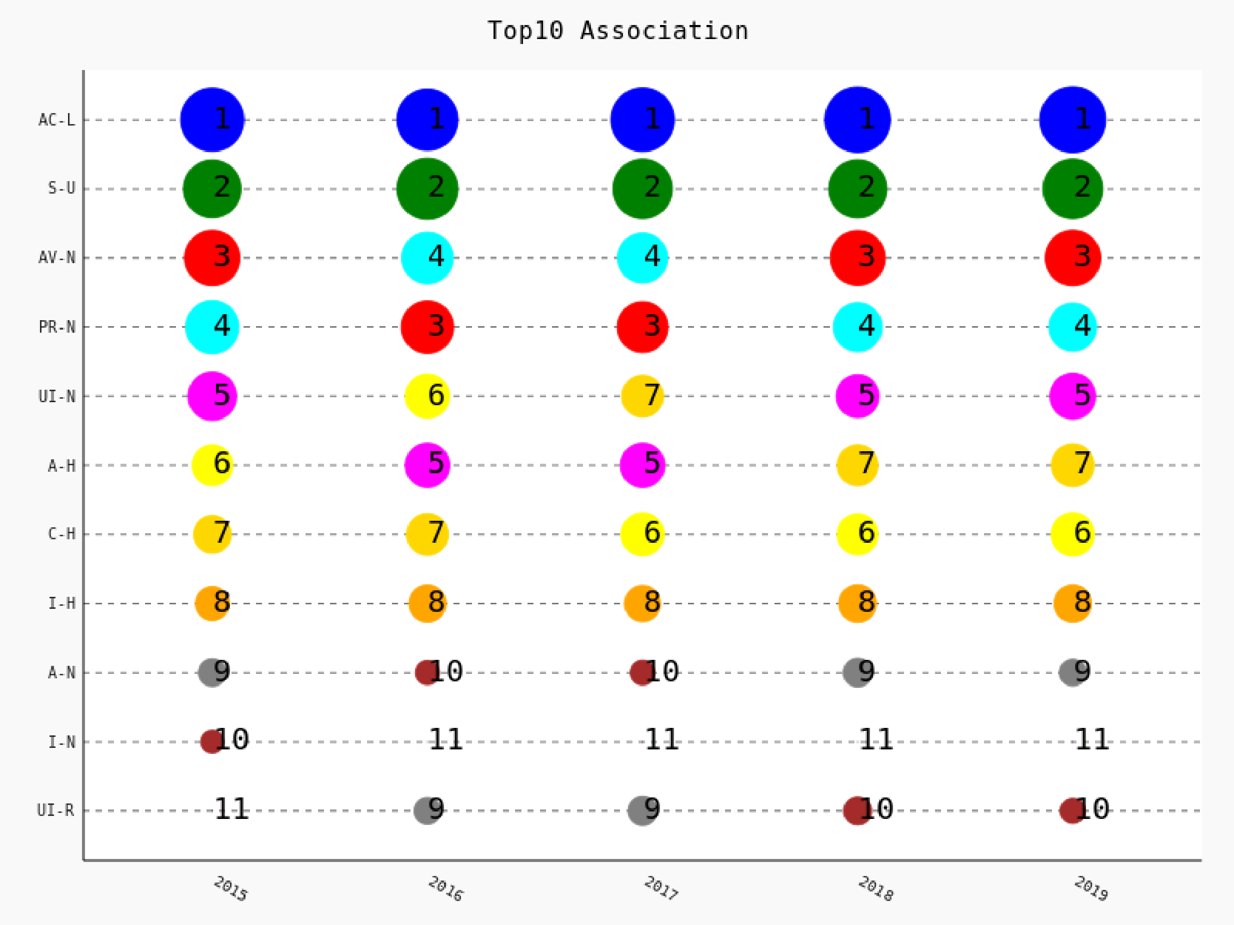


Fig. 3. CVSS v3 top 10 rankings

In conclusion, our data indicates that the vulnerability threat landscape has been dominated by a few vulnerability types and has not evolved over the years. The overwhelming majority of software vulnerabilities are exploitable over the network (i.e., remotely). The complexity to successfully exploit these vulnerabilities is dominantly low and very little authentication to the target victim is necessary for a successful attack. Moreover, most of the flaws require very limited interaction with users. The damage of these vulnerabilities has, however, mostly been confined within the scope of the compromised systems.

IV. Related Work

There are many efforts to understanding the software vulnerability landscape. These efforts include reports by security solutions vendors [18], [19], white papers from non-profits such as MITRE [20] and SANS [21], as well as academic papers. For CVSS, most studies focused on the aggregation equation that produces the CVSS numerical scores representing the severity of the vulnerability. Surprisingly, we found no studies on v3 despite its preponderance in commercial security software.

Reference [8] is among the first statistical studies of the CVSS scoring system. It evaluated v1 and proposed improvements that contributed to the release of v2. Our study considers both v2 and v3 (but does not try to improve on either). Relative to the statistical evaluation, we consider our paper as a continuation and update of the work in [8]. However, our work uses data from a much longer time period. It also goes one step further by analyzing association rules of vulnerability metrics. It is worth noting that there are similarities between the results of the two studies. For instance, both papers show the predominance of certain types of vulnerabilities. Our historical analysis (which was not performed in [8]) shows that this predominance is maintained over the years.

Reference [11] considers CVSS v1 and v2 and analyzes how effectively v2 addresses the deficiencies found in v1. It also identifies new deficiencies. In contrast, our motivation was to understand the threat landscape.

Reference [13] uses empirical data from an international cyber defense exercise to study how 18 security estimation metrics based on CVSS correlate with the actual *Time-To-Compromised* (TTC) of 34 successful attacks. This study uses TTC as a dependent variable to analyze how well different security estimation models involving CVSS are able to approximate the actual security of network systems. The results suggest that security modeling with CVSS data alone does not accurately portray the time-to-compromise of a system. This result questions the applicability of the CVSS numerical scoring equation. Our study focused on the raw CVSS vectors, which represent the actual experts' opinions about the vulnerabilities.

Reference [22] uses NVD data to study trends and patterns in software vulnerabilities in order to predict the time to next vulnerability for a given software application. Data mining techniques were used as prediction tools. The vulnerability features used to aid the prediction are the published time of each vulnerability and its version. We believe that these features are not sufficiently informative. Instead, we directly use the eight metrics from the CVSS base scores which constitute the best available information covering large multi-year sets of vulnerabilities.

Reference [23] also carried out a predictive study based on the NVD/CVSS and ExploitDB [24] data. Using the CVSS data, it attempts to answer two questions: *(1) Can we predict the time until a proof of concept exploit is developed based on the CVSS metrics? and (2) Are CVSS metrics populated in time to be used meaningfully for exploit delay prediction of CVEs?* The former is answered in the positive, while the latter is answered in the negative. While using the same datasets, our objective differs from that in [23]. We did not attempt to predict the threat landscape; we provide a thorough historical and statistical study of vulnerabilities for the last fifteen years.

The work in [25] is another assessment of CVSS. It evaluates the trustworthiness of CVSS by considering data found in five vulnerability databases: NVD, X-Force, OSVDB (Open Source Vulnerability Database), CERT-VN (Computer Emergency Response Team, Vulnerability Notes Database), and Cisco IntelliShield Alerts. It then uses a Bayesian model to study consistencies and differences. It concluded that CVSS is trustworthy and robust in the sense that most of the databases generally agree. This suggests that our focus on the NVD to study the threat landscape is justified: studies using data from the other databases will likely lead to the same conclusions.

All of the studies cited above are focused on v1 and v2. In our literature survey, we did not find a single study that uses the updated and significantly modified v3 to understand the software vulnerability landscape. We believe that the present paper is the first of this kind in doing so. Furthermore, our study is the first to use association rule mining and co-occurrence of vulnerability metrics' values in an attempt to characterize the software threat landscape.

V. Conclusion

Our data indicates that the vulnerability threat landscape for publicly disclosed vulnerabilities has been dominated by a few vulnerability types and has not significantly changed from 2005 to 2019. However, the underlying software flaw types that enable these vulnerabilities change dramatically from year to year (for example, see [26]). This means that many flaw types result in vulnerabilities with the same properties. This is bad news because it means, as a security community, it will be difficult to eliminate certain vulnerability types because they result from a plethora of underlying software flaw types.

Another concern is that the overwhelming majority of software vulnerabilities are exploitable over the network. When developing software, efforts should be made to reduce unnecessary connections, protect necessary ones, and require more authentication where possible to reduce attack surface area. Another significant issue is that most of the vulnerabilities require no sophistication to be exploited (but again this is hard to improve upon due to the many software flaw types that allow this).

These two factors together combined with the finding that most vulnerabilities require very limited interaction with users facilitates the widespread hacking occurring today. Often in security literature the human is cited as the weakest link. While certainly humans can be exploited, within the set of CVE type vulnerabilities, exploitation of humans plays a very minor role. Hence, although training humans might always help strengthen security, to obtain a better impact in this area, the priority should be shifted to correcting these constant vulnerabilities.

Overall, this study documents the security community's inability to eliminate any types of vulnerabilities through addressing the related software flaw types. In 15 years, the vulnerability landscape has not changed; through the lens of the metrics in this paper we are not making progress. Perhaps we as community need to "stop and think" about the ways we are developing software and/or the methods we use to identify vulnerabilities. The security community needs new approaches. We do not want to write this same paper 15 years from now showing that, once again, nothing has changed.

Overall, this study shows that either we (the community) are incapable of correcting the most common software flaws, or we are focusing on the wrong flaws. In either case, it seems to us that there is a need to "stop and think" about the ways we are developing software and/or the methods we use to identify vulnerabilities.

Acknowledgement

This work was partially accomplished under NIST (National Institute for Standards and Technology) Cooperative Agreement No.70NANB19H063 with Prometheus Computing, LLC. The authors would like to thank the NVD staff for their review and consideration of this work.

Any mention of commercial products or entities is for information only; it does not imply recommendation or endorsement.

References

[1] "National vulnerability database," 2020, accessed: 2020-01-10. [Online]. Available: https://https://nvd.nist.gov

[2] "Common vulnerability scoring system special interest group," accessed: 2019-12-10. [Online]. Available: https://www.first.org/cvss

[3] "Forum of incident response and security teams," accessed: 2020-01-10. [Online]. Available: https://www.first.org/

[4] M. Schiffman, A. Wright, D. Ahmad, and G. Eschelbeck, "The common vulnerability scoring system," *National Infrastructure Advisory Council, Vulnerability Disclosure Working Group, Vulnerability Scoring Subgroup*, 2004.

[5] P. Mell, K. Scarfone, and S. Romanosky, “A complete guide to the common vulnerability scoring system version 2.0,” in *Published by FIRST-Forum of Incident Response and Security Teams*, vol. 1, 2007, p. 23.

[6] “Common vulnerability scoring system v3.0: Specification document,” accessed: 2020-2-5. [Online]. Available: https://www.first.org/cvss/v3.0/specification-document

[7] “Common vulnerability scoring system v3.1: Specification document,” accessed: 2020-2-5. [Online]. Available: https://www.first.org/cvss/v3.1/specification-document

[8] P. Mell and K. Scarfone, “Improving the common vulnerability scoring system,” *IET Information Security*, vol. 1, no. 3, pp. 119–127, 2007.

[9] P. Mell, K. Scarfone, and S. Romanosky, “Common vulnerability scoring system,” *IEEE Security & Privacy*, vol. 4, no. 6, pp. 85–89, 2006.

[10] H. Holm and K. K. Afridi, “An expert-based investigation of the common vulnerability scoring system,” *Computers & Security*, vol. 53, pp. 18–30, 2015.

[11] K. Scarfone and P. Mell, “An analysis of cvss version 2 vulnerability scoring,” in *Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement*. IEEE Computer Society, 2009, pp. 516–525.

[12] R. Wang, L. Gao, Q. Sun, and D. Sun, “An improved cvss-based vulnerability scoring mechanism,” in *2011 Third International Conference on Multimedia Information Networking and Security*. IEEE, 2011, pp. 352–355.

[13] H. Holm, M. Ekstedt, and D. Andersson, “Empirical analysis of system-level vulnerability metrics through actual attacks,” *IEEE Transactions on dependable and secure computing*, vol. 9, no. 6, pp. 825–837, 2012.

[14] D. W. Baker, S. M. Christey, W. H. Hill, and D. E. Mann, “The development of a common enumeration of vulnerabilities and exposures,” in *Recent Advances in Intrusion Detection*, vol. 7, 1999, p. 9.

[15] “Common vulnerabilities and exposures,” 2020, accessed: 2020-2-5. [Online]. Available: https://cve.mitre.org

[16] A. Gueye and P. Mell, “A historical and statistical studyof the software vulnerability landscape,” 2021.

[17] NVD, “private communication,” Mar. 2019.

[18] Symantec, “2019 internet security threat report,” 2020, accessed: 2020-02-01. [Online]. Available: https://www.symantec.com/content/dam/symantec/docs/reports/istr-24-2019-en.pdf

[19] McAfee, “Mcafee labs 2019 threats predictions report,” 2020, accessed: 2020-02-01. [Online]. Available: https://www.mcafee.com/blogs/other-blogs/mcafee-labs/mcafee-labs-2019-threats-predictions/

[20] MITRE, “2019 cwe top 25 most dangerous software errors,” 2020, accessed: 2020-02-01. [Online]. Available: https://cwe.mitre.org/top25/archive/2019/2019_cwe_top25.html

[21] SANS, “2020 sans cyber threat intelligence (cti) survey,” 2020, accessed: 2020-02-01. [Online]. Available: https://www.sans.org/reading-room/whitepapers/analyst/2020-cyber-threat-intelligence-cti-survey-39395

[22] S. Zhang, D. Caragea, and X. Ou, “An empirical study on using the national vulnerability database to predict software vulnerabilities,” in *International Conference on Database and Expert Systems Applications*. Springer, 2011, pp. 217–231.

[23] Y. Y. A. Feutrill, D. Ranathunga and M. Roughan, “The effect of common vulnerability scoring system metrics on vulnerability exploit delay,” in *2018 Sixth International Symposium on Computing and Networking (CANDAR), Takayama*, 2018, pp. 1–10.

[24] “Exploit database,” 2020, accessed: 2020-02-01. [Online]. Available: https://www.exploit-db.com/

[25] P. Johnson, R. Lagerstrom, M. Ekstedt, and U. Franke, “Can the common vulnerability scoring system be trusted? a bayesian analysis,” *IEEE Transactions on Dependable and Secure Computing*, 2016.

[26] “National vulnerability database, cwe over time,” 2019, accessed: 2019-12-10. [Online]. Available: https://nvd.nist.gov/general/visualizations/vulnerability-visualizations/cwe-over-time

Conceptualization of A GDPR-Mining Blockchain-Based Auditor: A Systematic Review

Gholamhossein Kazemi
Department of Business, Marketing and Law
University of South-Eastern Norway
Hønefoss, Norway
Email: 238834@usn.no

Shegaw Anagaw Mengiste
Department of Business, History and Social Sciences
University of South-Eastern Norway
Vestfold, Norway
Email: Shegaw.mengiste@usn.no

***Abstract*—This paper is a systematic literature review of the compliance of blockchain with the General Data Protection Regulation act. Although there are contradictory opinions about the compliance of blockchain with General Data Protection Regulation amongst different researchers, in this paper, we conduct a systematic literature review on the topic to get a perspective on previous studies and models to build a conceptual blockchain-based General Data Protection Regulation-mining two-way monetized auditor design upon existing solutions and models for an interactive software auditing the transactions between the data subjects and third parties. This review aims to answer the dilemma of the applicability of blockchain in auditing the transactions between the data subjects and data processors in the General Data Protection Regulation framework. Moreover, this paper discusses the implications and limitations and paves the path for future studies to elaborate on the concept.**

Keywords-blockchain; GDPR; consensus; auditor.

I. INTRODUCTION

Since the emergence of Bitcoin in 2009, based on Blockchain technology, a vast group of academic, industrial, and business innovators have become more and more attracted to using blockchain technology for their purposes such as researches, literature reviews, secure contracts, information sharing, and digital transactions due to its immutable, transparent, secure, and trustworthy characteristics [1][2].

Accordingly, blockchain could be the best solution for privacy protection and data processing transparency regarding its compatibility with the General Data Protection Regulation (GDPR) act that is effective since May 25th 2018 across Europe; however, there are contradictory opinions around its compatibility among scholars [3][4][5]. Numerous pieces of literature introduced the disruptive capabilities of blockchain as the most important revolutionizing invention after the Internet itself, considering its distributed consensus model [6].

A systematic review of more than 260 scholarly articles about blockchain applications from 2014 to the first quarter of 2018 illustrates that only 24 had focused on the privacy and security area, with more than 1000 percent growth in the second half [7]. Hence, regarding the exponential soar in this field, and considering the launch of GDPR in the second quarter of 2018, this study scrutinizes the contrasting notions about the trending less-investigated concept of blockchain-GDPR harmony and develops a conceptual model for an interactive software auditing the interactions between the data subjects and third-party data processors under the supervision of GDPR issuing-parties based on the previous scholarly designs [5]. Hence, the research question is whether a blockchain-based platform is capable of auditing the transactions between the data subjects and third-party data processors in the framework of GDPR or not?

In this regard, this paper analyzes 49 articles and proceedings from 2016 to 2020 to pinpoint the applications and challenges of blockchain and GDPR compatibility of blockchain technologies. This review consists of two main time periods, before the launch of GDPR and after the launch of GDPR, and classifies the applications of GDPR in the mentioned periods to identify the gap for the inception of a blockchain-based GDPR auditing moderator, also at the same time, underpins the issues of implementation of such an application. Moreover, this paper discusses the implications and limitations and paves the path for future studies to elaborate on the concept. This paper contains five parts including introduction as Section I, literature review as Section II covering subtitles of review process, constructs, applications, and compliance, results as Section III, discussion as Section IV, and conclusion as Section V.

II. LITERATURE REVIEW

This study seeks the answer to the dilemma of applicability of blockchain in auditing the transactions between the data subjects and data processors in the GDPR framework that is provided by the EU and imposes strong obligations regarding security and privacy to all of the organizations around the world that collect or process any data related to the people in the EU [3].

Moreover, the findings of this study are related to the context of GDPR articles, and the level of analysis is the applications of blockchain technology, which are highly dependent on its components, mechanisms, and consensuses [8]. Consequently, the low diversity of the mechanisms of blockchain and the translation of the GDPR articles into logical machine algorithms are the limitations of this conceptual model that need further development in future studies. Finally, the outcome of this study serves the interests of data subjects, third-party data processors, and the auditing organization for the GDPR acts. All the steps of this systematic review are explained in the following sections and depicted in Figure 1.

A. *Review Process*

To attain a holistic approach towards the proposed model, a preliminary search from 2016 to 2020 with the keywords "Blockchain Applications", "Blockchain Issues", "Blockchain Security", "Blockchain Privacy", and "Blockchain and GDPR" was conducted through Web of Science, Scimago Journal and Country Rank, and Google Scholar to help define the concepts, categorize the applications of blockchain, and assess its compatibility with GDPR. As a preliminary result, 89 articles and proceedings were selected for further investigations while after filtration on source journals and proceedings reliability, H5 index rate, citations rate, abstract and keywords relevancy, and result and conclusion validity and novelty, 40 articles and proceedings were found ineligible and 49 articles and proceedings were selected for the second round of filtration, as illustrated in Table I. Filtered articles and proceedings either have a high citation, a high H5 index from the publisher journal or proceeding, or a valuable content due to its novelty of publishing date. The initial inclusion and exclusion criteria of this systematic review are, respectively, the publication year of the study being between 2016 to 2020, the credibility of the study assessed by the citation rate of the study and H5 index rate of the publisher, the relevancy of the study evaluated by scanning the keywords and abstracts, and the novelty of the results and conclusions. Regarding this criteria, the selected articles and proceedings were assigned into the subcategories of either supportive or key articles and proceedings, resulting in 21 supportive and 28 key articles and proceedings.

TABLE I. DISPERSAL OF THE FOUNDED ARTICLES BY YEAR, H5 INDEX, CITATION, AND KEYWORDS

Year	No of Articles and Proceedings	Min-Max H5 Index	Min-Max Citation	Keywords
2016	5	0-300	118-1098	Applications, Crypto, Issues, Technology
2017	6	0-231	1-518	Applications, Concepts, Consensuses, Crypto, Issues, Privacy Smart Contracts
2018	10	0-231	0-1159	Applications Concept, Consensuses, Crypto, GDPR, Issues, Privacy, Security, Smart Contract
2019	19	0-169	1-339	Applications, Consensuses, Crypto, GDPR, Issues, Privacy Security, Smart Contract
2020	9	0-125	0-545	Applications, Consensuses, Crypto, GDPR, Issues, Privacy, Security, Smart Contract
Overall	49	0-300	0-1159	

The second round of filtration consists of the classification of the 28 key articles and proceedings based on whether they contain information concerning the definition of constructs, the applications of blockchain, or the compatibility of the blockchain and GDPR, resulting in 7 key articles and proceedings concerning constructs, 17 applications, and 4 compatibility aspects. Finally, a three-step literature review reveals the definitions of the constructs, classifies the applications of blockchain, and demonstrates the compatibility of blockchain and GDPR. In each step, the minimum quantitative obligations are a minimum average citation of 150 or a minimum average H5 index of 65, and the minimum qualitative requirement is the verification of a supportive supervisory team including researchers with relevant research experience.

B. *Constructs*

In the first step of this systematic literature review, this paper derives the definitions of the foundational concepts and mechanisms embedded in blockchain technology and GDPR by reviewing 7 key articles and proceedings from 2016 to 2020 to integrate the notions about the basis of the concepts. These articles and proceedings have an average citation of 150 and an average H5 index of 65 with a citation range of 1 to 518 and an H5 index rate of 19 to 112.

GDPR is one of the largest and most difficult regulations in data privacy history issued by the European Union (EU) party across Europe with data subjects' consent centricity. This regulation applies to all the data processors worldwide offering personal data-related goods and services to the citizens of the EU and the data processors located in the EU providing services for the rest of the world. Besides, the data subject's consent should be withdrawable, the data should be removable, and the processing purpose of the third parties should be clear and accessible to the data subjects [9]. Moreover, one of the applications of the GDPR is the compensation of the data subjects suffering from data privacy violations. This reimbursement takes place by fining the data processors breaking the rules of GDPR, although this is only one-way monetized [3].

Blockchain technology is a synthesis of techniques of cryptography, algorithms, distributed consensuses, immutable databases, and distributed peer to peer networks that propagate blocks containing Hash as the modification notifier and function propagator, Timestamp as the time recorder of the transactions, and data subblocks containing specific programmed data [4][8].

Consensus algorithm enables the establishment of a mutual trust between the users of a blockchain network without any need for an administrative party to verify the transaction between them. In other words, the "consensus function is a mechanism that makes all blockchain nodes have

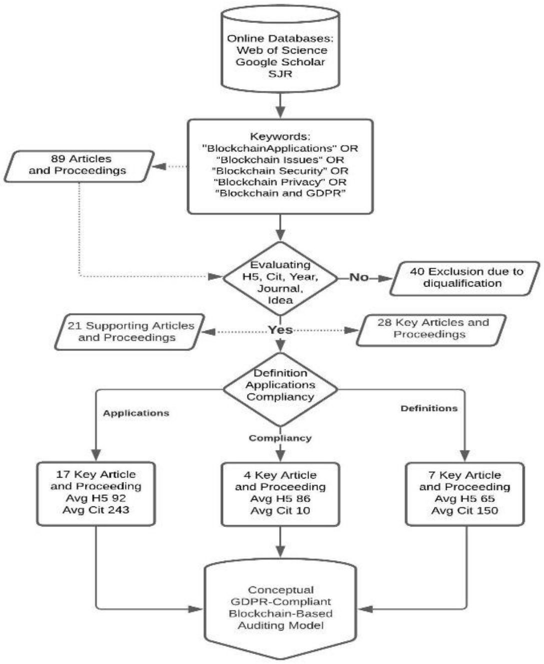

Online Databases:
Web of Science
Google Scholar
SJR
Keywords:
"BlockchainApplications" OR
"Blockchain Issues" OR
"Blockchain Security" OR
"Blockchain Privacy" OR
"Blockchain and GDPR"
89 Articles and Proceedings
Evaluating H5, Cit, Year, Journal, Idea
No
40 Exclusion due to diqualification
Yes
21 Supporting Articles and Proceedings
28 Key Articles and Proceedings
Definition Applications Compliancy
Applications
Compliancy
Definitions
17 Key Article and Proceeding Avg H5 92 Avg Cit 243
4 Key Article and Proceeding Avg H5 86 Avg Cit 10
7 Key Article and Proceeding Avg H5 65 Avg Cit 150
Conceptual GDPR-Compliant Blockchain-Based Auditing Model

Figure 1. Systematic Review Process.

an agreement in the same message" [10].

Some consensus algorithms are like Proof of Work (PoW) that requires solving a complicated computational process to ensure authentication and verifiability to mine a block of transactions in a blockchain [7][11], Proof of Stake (PoS), which validates the users that present their holdings to generate the next block while another version of PoS is called Delegated Proof of Stake (DPoS), which aims at completing a distributed consensus in the system [7][11], and Zero-Knowledge Proof (ZKP) that in private transactions makes the verifier believe that the target information exists in the transaction, although it does not reveal the real information [11].

Smart contract is an agreement between doubtful members, implemented by the consensus mechanisms in which trusted transactions received by the blockchain can call the contract's public methods to use its data for processing [12].

C. *Applications of Blockchain*

In the second step, this review divides the target period of the investigation into the before and after the launch of GDPR, which is 2016 to the first quarter of 2018, and the second quarter of 2018 to 2020, and scrutinizes the applications of blockchain in scholarly articles and provides a classification for the covered fields. The classification and dispersal of the applications of blockchain are shown in Table II. This table shows the number of articles and proceedings in each period and the applications that each has mentioned.

In this step, 6 articles from the first period with an average citation of 462 and an average H5 index of 87 were scanned that had a citation range of 11 to 1098 and an H5 index range

TABLE II. APPLICATIONS OF BLOCKCHAIN DISPERSAL AND CLASSIFICATION BY H5 INDEX, CITATION, AND YEAR

Period	No of Articles and Proceedings	Applications	H5 Index	Citation	Year
2016 to 1st quarter of 2018	6	Healthcare, Privacy and Security, Finance, Database, IoT, Other	31	11	2018
		Finance, Privacy and Security, IoT, Health care, Other	35	518	2017
		Healthcare, IoT, Finance, Privacy and Security, Other	19	97	2018
		Finance, Other	231	360	2017
		Finance, Privacy and Security, Other	203	688	2016
		Privacy and Security, Finance, IoT, Other	0	1098	2016
Average			87	462	
2nd quarter of 2018 to 2020	11	Privacy and Security, Database, Healthcare, Other	148	26	2019
		Healthcare, Finance, Privacy and Security, Other	169	95	2019
		Finance, Privacy and Security, Healthcare, Database, IoT	56	395	2019
		Healthcare, IoT, Other	67	13	2020
		Finance, Privacy and Security, Other	24	68	2018
		Other	125	46	2019
		Finance, Healthcare, Privacy and Security, Other	35	0	2020
		Finance, Privacy and Security, Other	174	377	2019
		Database, Other	86	201	2019
		Finance, Database, Privacy and Security, Other	99	128	2019
		Health care, Finance, IoT, Privacy and Security, Other	67	13	2019
Average			95	124	
Overall			92	243	

of 0 to 231, also 11 articles from the second period with an average citation of 124 and an average H5 index of 95 were scanned that had a citation range of 24 to 174 and H5 index range of 0 to 395. The mentioned table classifies the applications of blockchain into Healthcare, Finance, Database, Privacy and Security, Internet of Things (IoT), and others, and illustrates that in these 5 years 17 key articles have mentioned these applications 62 times, although none has mentioned a blockchain-based application for the audition of the transactions between the data subjects and third parties in the framework of GDPR, even after the launch of GDPR.

On the other hand, some articles have discussed the compliance of other blockchain-based applications with the GDPR act that will be investigated in the next section.

D. *Compliance of Blockchain With GDPR*

In the third step, this review investigates the compliance of the concept of blockchain with the GDPR act. To achieve the result, the contradictory notions are extracted from 4 key scholarly articles and proceedings with an average citation of 10 ranging from 1 to 28 and an average H5 index of 86 ranging from 77 to 112. The dispersal of GDPR inconsistencies of blockchain and their solutions are illustrated in Table III by classification of issues, solutions, H5 index, citation, and year.

On one hand, data immutability in blockchain technology is in contrast with the GDPR act that entitles the users to delete their data, on the other hand, one solution to tackle the crisis of data immutability in the blockchain is using techniques like Accenture that lets a trusted party alter the data block, Monero that makes the data subjects untraceable, and Hyperledger that transforms blockchain to a code executable distributed computer [13]. Moreover, a Hyperledger Fabric permissioned blockchain can use smart contracts to detect trusted parties, an off-chain storage method to reduce data leakage, and eXtensible Access Control Markup Language to impose governance measures to tackle the inconsistency of blockchain and GDPR in a blockchain-based personal data management application [4].

TABLE III. DISPERSAL AND CLASSIFICATION OF BLOCKCHAIN INCONSISTENCIES WITH GDPR AND THEIR SOLUTIONS BY H5 INDEX, CITATION, AND YEAR

No of Articles and Proceedings	**GDPR Inconsistency**	**Solution**	**H5 Index**	**Citation**	**Year**
4	Data Erasure, Privacy	Smart Contract, Monero, Accenture Altering Technique, Hyperledger, Etherium	77	7	2019
	Data Erasure, Privacy, Data Governance	Smart Contract, Hyperledger, Etherium, Off-Chain Storage, XACML, SecKit	112	1	2019
	Data Erasure, Privacy	Layered Architecture, Off-Chain Storage, Private Blockchain, Data Depersonalization	77	2	2020
	Data Erasure, Privacy, Data Governance	Layered Blockchain, Smart Contract, Digital Verification, Off-Chain Storage	77	28	2019
Average			86	10	

Another prototype overcoming compliance issues is the German Asylum case that uses the layered architecture of information access and storage, private blockchain, and data depersonalization methods to harmonize the ongoing procedures with the GDPR [14].

In another example, Personal Data And Identity Management blockchain-based application, with a human-centric approach, designs layered blockchains with smart contracts, permissioned access, digital identity verification, and off-chain storage for consent and identity management and data monetization [9].

III. RESULTS

This literature review reveals that although there has been a remarkable increase in the number of scholarly articles exploring the applications of blockchain in the privacy and security area before and after the lunch of GDPR, there is still a gap in unfolding high potential capabilities of blockchain as a GDPR-compatible technology, which is capable of providing the basis for the audition of the transactions between the data subjects and data processors. As disclosed previously, 17 articles and proceedings have mentioned the applications of blockchain 62 times from 2016 to 2020 while only 4 articles and proceedings have investigated its applications in GDPR-related topics like human-centric data management services or one-way monetized personal data management services [4][9][13]. Furthermore, none has indicated blockchain's capability as a basis for a two-way monetized GDPR-mining auditing platform.

After extraction of the concepts and constructs of GDPR and blockchain from the literature, classification of the explored applications of blockchain, and investigation of technical compatibility of GDPR and blockchain, this study explicates that there might be illusive inconsistencies in the definitions of GDPR and blockchain at a superficial level; however, at a technical level techniques and technologies like Smart Contracts, Monero, Accenture, Hyperledger, Off-Chain Storage, etc. reinforce the unseen bonds between the interrelated motifs of GDPR and blockchain [4][14].

Consequently, after clarification of the compatibility of GDPR and blockchain regarding the research's question, and after exploration of the previously proposed solutions and models, this study aims to fill the gap with a conceptual two-way monetized GDPR-mining blockchain-based auditing platform to fulfill the necessity of an effective transaction auditor platform as a supervisory authority, which is capable of fining data privacy violators and rewarding trader data subjects.

IV. DISCUSSION

After three rounds of systematic literature review and analysis containing definition extraction, application gap detection, and blockchain-GDPR compatibility assessment, this study develops a conceptual model based on the previous prominent GDPR-compliant blockchain-based data management applications and builds up the GDPR article mining concept and two-way monetizing contracts upon previous models.

Previous models introduced in the reviewed papers were designed to enhance the security and privacy of managing personal data in the framework of GDPR with the help of blockchain technology, also one-way monetization is mentioned in one of the previously designed models [4][5][9].

Although significant efforts have been made at a technical level for the management of personal data, there is still a need for an auditing platform capable of two-way monetized audition accompanied by the feature of the GDPR-mining concept. Therefore, this study aims to build upon previous models and conceptualize a two-way monetized auditing platform in which data processors can mine the GDPR acts as nodes in the blockchain. In this conceptual three-layered blockchain-based model, an issuing party acts as a supervisory authority that stores, audits, and monetizes the transactions between the data subjects and the third parties based on smart contracts. Figure 2 illustrates the model.

This model consists of a public blockchain for the registration and credit evaluation of the third parties that permits all the data processors to register as a verified member on the blockchain and to interact with the data subjects in order to build up an agreement with them in the framework of a smart contract for the monetization of their relation regarding data processing and data trading [5][9]. This blockchain evaluates the data processors based on PoW consensus after completion of each cycle of transaction that goes through the three-layered blockchain and comes back with the result of the process. Data processors can mine the GDPR nodes and earn value and credit as long as they prove more and more nonconflicting results with the GDPR nodes [7][11].

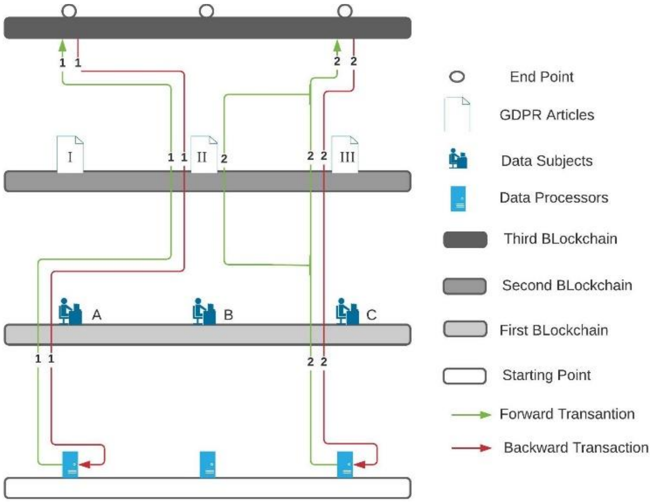


Figure 2. Conceptual model of the three-layered blockchain-based auditor.

Also, a private blockchain of data subjects and their customized governance policies for communication with the third parties is an intermediate layer that stores all the information of the data subjects in blocks of data using consensus algorithms and techniques of data propagation and data alteration in order to empower the data subjects to chose how they want to share their blocks of data and when they want to share or withdraw their blocks of data [4][7][10][11][13]. In this blockchain, smart contracts clarify the agreement of monetization and data accessibility. In this regard, data processors send their request of fetching data with the transparently defined act of GDPR, which is needed to be taken into account, in the form of a PoS or DPoS consensuses. As long as data processors share their act, they can fetch information and mine GDPR nodes to elevate their credits [7][11]. Also, the ZKP consensus ensures the availability of information on the data subject side and the availability of funds on the data processor side for further monetization in the form of data trading or violation fining [11]. Monetization is based on the inputs of involved parties in the agreement of the smart contract.

The third layer is a consortium blockchain of machine-translated GDPR article nodes in which each node presents one specific article of GDPR and its requirements and relations. Requirements obligate the data processors to process the data in the framework of GDPR, and the relations present the possible connection of the different acts of GDPR as nodes in the blockchain. Each time a data processor reaches the information of a data subject through a smart contract, the transaction of fetching and processing of information goes through the GDPR layer for further evaluation and audition of the process. Moreover, a backward transaction containing processed information of the data subject comes back through the GDPR layer and notifies the data subjects about the way their information is being used. This backward transaction helps the evaluation of the third parties in an assessment cycle while the third parties can either enhance their credit as they mine more and more GDPR articles nodes or lose credit due to GDPR violations.

Finally, all the transactions and information are stored on an off-chain server of the auditing issuing party that enables the issuing party to trace the footprints and audit the transactions based on the agreements between the involved parties to either reward the data subjects for selling the data or fine the third parties in case of GDPR-conflicting transactions [4][9]. For instance, as illustrated in Figure 2, at the starting point, data processor number 1 registers on the first blockchain via transaction 1 and requests the establishment of a smart contract with data subject A. After initiation of the monetization agreement and clarification of the act, it fetches the demanded data from the second blockchain and mines the related GDPR act number II. Eventually, after the process of the data at the end point, a backward transaction containing the processed data travels back to the start point, where a supervisory authority stores all the information of the transaction on the off-chain storage and audits the transaction based on the smart contracts in order to validate the GDPR-act mining of the data processor and monetize the transaction. Similarly, data processor number 2 goes through the same procedure via transaction 2, however it mines two GDPR acts due to the relevancy of its purpose to those acts.

This is an early-stage conceptual design and needs further development due to its technical and practical limitations like the translation of GDPR acts into machine algorithms adjustable in the framework of blockchain, unavailability of customized blockchain mechanisms, and possible refusal of the involved parties for the implementation of such a platform.

V. Conclusion and Future Work

To recapitulate, in three-rounds of systematic analysis, this paper extracts the proper definitions for the understanding of the concepts of blockchain and GDPR, classifies the applications of blockchain, and demonstrates that neither before nor after the launch of GDPR no scholarly article has mentioned the application of blockchain in auditing and monetizing the transactions between the third parties and data subjects. However, after the launch of GDPR, some scholars have investigated the inconsistency of blockchain-based applications with GDPR acts and proposed designed solutions.

Finally, this study builds upon those designs and proposes an interactive conceptual GDPR-mining blockchain-based auditing model capable of GDPR node mining and two-may monetizing. This is an initial conceptual design and further investigation regarding practicality of the model needs to be done, and developments need to be made in the future.

Acknowledgment

This work would not have been possible without the help of the faculty members of the University of South-Eastern Norway. Hence, I would like to thank them all.

References

[1] D. Dasgupta, J. M. Shrein, and K. D. Gupta, “A survey of blockchain from security perspective,” Journal of Banking and Financial Technology, no. 3, pp. 1-17, 2019, doi:10.1007/s42786-018-00002-6.

[2] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen, “A survey on the security of blockchain systems,” Future Generation Computer Systems, no. 107, pp. 841-853, 2020, doi:10.1016/j.future.2017.08.020.

[3] B. Wolford. GDPR.EU: What is GDPR, the EU’s new data protection law?. [Online]. [Retrieved: March, 2021] Available from: https://gdpr.eu/what-is-gdpr/

[4] N. B. Truong, K. Sun, G. M. Lee, and Y. Guo, “GDPR-compliant personal data management: a blockchain-based solution,” IEEE Transactions on Information Forensics and Security, vol. 15, pp. 1746-1761, 2019, doi:10.1109/TIFS.2019.2948287.

[5] C. Wirth and M. Kolain, “Privacy by blockchain design: a blockchain-enabled GDPR-compliant approach for handling personal data,” Reports of The European Society for Socially Embedded Technologies (EUSSET), vol. 6, no. 2, 2018, doi:10.18420/blockchain2018_03.

[6] M. Crosby, Nachiappen, P. Pattanayak, S. Verma, and V. Kalyanaraman, “Blockchain technology: beyond Bitcoin,” Applied Innovation Review, no. 2, 2016. [Online]. [Retrieved: March, 2021] https://j2-capital.com/wp-content/uploads/2017/11/AIR-2016-Blockchain.pdf

[7] F. Casino, T. K. Dasaklis, and C. Patsakis, “A systematic literature review of blockchain-based applications: current status, classification, and open issues,” Telematics and Informatics, no. 36, pp. 55-81, 2019, doi:10.1016/j.tele.2018.11.006.

[8] A. P. Joshi, M. Han, and Y. Wang, “A survey on security and privacy issues of blockchain technology,” Mathematical Foundations of Computing, vol. 2. no 1, pp. 121-147, 2018, doi:10.3934/mfc.2018007.

[9] B. Faber, G. Michelet, N. Weidmann, R. R. Mukkamala, and R. Vatrapu, “BPDIMS: a blockchain-based personal data and identity management system,” The 52nd Hawaii International Conference On System Sciences (HICSS 2019), Hawaii, 2019, pp. 6855-6864, ISBN: 978-0-9981331-2-6. [Retrieved: March, 2021] http://128.171.57.22/bitstream/10125/60121/0681.pdf

[10] I. C. Lin and T. C. Liao, “A survey of blockchain security issues and challenges,” International Journal of Network Security, vol. 19, no. 5, pp. 653-659, 2017, doi:10.6633%2fIJNS.201709.19(5).01.

[11] E. Zaghloul, T. Li, M. W. Mutka, and J. Ren, “Bitcoin and blockchain: security and privacy,” IEEE INTERNET OF THINGS JOURNAL, vol. 7, no. 10, pp. 10288-10313, 2020, doi:10.1109/JIOT.2020.3004273.

[12] A. Ghosh, S. Gupta, A. Dua, and N. Kumar, “Security of cryptocurrencies in blockchain technology: state-of-art, challenges, and future prospects,” Journal of Network and Computer Applications, vol. 163, 2020, doi:10.1016/j.jnca.2020.102635.

[13] S. Farshid, A. Reitz, and P. Robbach, “Design of a forgetting blockchain: a possible way to accomplish GDPR compatibility,” The 52nd Hawaii International Conference On System Sciences (HICSS 2019), Hawaii, 2019, pp. 7087-7095, ISBN: 978-0-9981331-2-6. [Retrieved: March, 2021] http://128.171.57.22/bitstream/10125/60145/0705.pdf

[14] F. Goggenmos, A. Wenninger, A. Rieger, G. Fridgen, and J. Lockl, “How to design a GDPR-compliant blockchain solution for cross-organizational workflow management: evidence from the German Asylum procedure,” The 53rd Hawaii International Conference On System Sciences (HICSS 2020), Hawaii, 2020, pp. 4023-4032, ISBN: 978-0-9981331-3-3. [Retrieved: March, 2021] http://128.171.57.22/bitstream/10125/64234/0397.pdf

[15] R. Stephen, and A. Alex, “A review on blockchain security,” The International Conference on Recent Advancement and Effectual Researches in Engineering Science and Technology (RAEREST 2018), Kerala State, India, 2018, doi:10.1088/1757-899X/396/1/012030.

[16] J. Xie, H. Tang, T. Huang, F. R. Yu, R. Xie, J. Liu, and Y. Liu “A survey of blockchain technology applied to smart cities: research issues and challenges,” IEEE Communications Surveys and Tutorials, vol. 21, no. 23, pp. 2794-2830, 2019, doi:10.1109/COMST.2019.2899617.

[17] A. P. Joshi, M. Han, and Y. Wang, “A survey on security and privacy issues of blockchain technology,” Mathematical Foundations of Computing, vol. 1, no. 2, pp. 121-147, 2018. [Retrieved: March, 2021] http://www.aimsciences.org/article/doi/10.3934/mfc.2018007

[18] P. Tasatanattakool and C. Techapanupreeda, “Blockchain: challenges and applications,” The 32nd International Conference on Information Networking (ICOIN 2018), Chiang Mai, Thailand, 2018, pp. 473-475. [Retrieved: March, 2021] https://www.researchgate.net/profile/Chian-Techapanupreeda/publication/324725048_Blockchain_Challenges_and_applications/links/5d2ec2d392851cf4408a852c/Blockchain-Challenges-and-applications.pdf

[19] S.Singh and N. Singh, “ Blockchain: future of financial and cyber security,” The 2nd International Conference on Contemporary Computing and Informatics (IC3I 2016), Noida, India, 2016, ISBN: 978-1-5090-5256-1. [Retrieved: March, 2021] https://www.researchgate.net/deref/http%3A%2F%2Fdx.doi.org%2F10.1109%2FIC3I.2016.7918009

[20] L. Moerel, “Blockchain & data protection…and why they are not on a collision course,” European Review of Private Law, vol. 26, no. 6, pp. 825-851, 2018. [Retrieved: March, 2021] https://media2.mofo.com/documents/191019-blockchain-data-protection.pdf

[21] A. Dorri, M. Steger, S. S. Kanhere, and R. Jurdak, “Blockchain: A distributed solution to automative security and privacy,” IEEE Communications Magazine, vol. 55, no. 12, pp. 119-125, 2017, doi:10.1109/MCOM.2017.1700879.

[22] F. Zemler and M. Westner, “Blockchain and GDPR: application scenarios and compliance requirements,” The Portland International Center for Management of Engineering and Technology Conference on Technology Management in the World of Intelligent Systems (PICMET 2019), Portland, USA, 2019, ISBN: 978-1-890843-39-7, doi:10.23919/PICMET.2019.8893923.

[23] M. Pournader, Y. Shi, S. Seuring, and S. C. L. Koh, “Blockchain applications in supply chain, transport and logistics: a systematic review of the literature,” International Journal of Production Research, vol. 58, no. 7, pp. 2063-2081, 2020, doi:10.1080/00207543.2019.1650976.

[24] S. Underwood, “Blockchain beyond Bitcoin,” Communications of the ACM, vol. 59, no. 11, pp. 15-17, 2016, doi:10.1145/2994581.

[25] N. Guar, “Blockchain challenges in adoption,” Managerial Finance, vol. 46, no. 6, pp. 849-858, 2020, doi:10.1108/MF-07-2019-0328

[26] L. W. Cong and Z. He, “Blockchain disruption and smart contracts,” The Review of Financial Studies, vol. 32, no. 5, pp. 1754-1797, 2019, doi:10.1093/rfs/hhz007.

[27] U. Roth, “Blockchain ensures transparency in personal data usage: being ready for the new EU General Data Protection Regulation,” European Research Consortium for Informatics and Mathematics News, no. 110, pp. 32-33, 2017. [Retrieved: March, 2021] https://ercim-news.ercim.eu/images/stories/EN110/EN110-web.pdf

[28] K. Salah, M. H. U. Rehman, N. Nizamuddin, and A. A. Fuqaha, “Blockchain for AI: review and open research challenges,” IEEE Access, vol. 7, pp. 10127-10149, 2019, doi:10.1109/ACCESS.2018.2890507.

[29] L. Hughes, Y. K. Dwivedi, S. K. Misra, N. P. Rana, V. Raghavan, and V. Akella, “Blockchain research, practice and policy: applications, benefits, limitations, emerging research themes and research agenda,” International Journal of Information Management, vol. 49, pp. 114-129, 2019, doi:10.1016/j.ijinfomgt.2019.02.005.

[30] J. H. Park and J. H. Park, “Blockchain security in cloud computing: use cases, challenges, and solutions,” Symmetry, vol. 9, no. 8, 2017, doi:10.3390/sym9080164.

[31] B. K. Mohanta, D. Jena, S. S. Panda, and S. Sobhanayak, “Blockchain technology: a survey on applications and security privacy and challenges,” Internet of Things, vol. 8, 2019, doi:10.1016/j.iot.2019.100107.

[32] G. Drosatos, A. Kritsas, C. Ilioudis, A. Papanikolaou, and A. P. Filippidis, “A blockchain-based platform for consent management of personal data processing in the IoT ecosystem,” Security and Communication Networks, vol. 2019, 2019, doi:10.1155/2019/1431578.

[33] C. Lima, “Blockchain-GDPR privacy by design: how decentralized blockchain internet will comply with GDPR data privacy.” Claudio Lima. [Online]. [Retrieved: March, 2021] https://blockchain.ieee.org/images/files/pdf/blockchain-gdpr-privacy-by-design.pdf

[34] V. Gramoli, “From blockchain consensus back to Byzantine consensus,” Future Generation Computer Systems, vol. 107, pp. 760-769, 2020, doi:10.1016/j.future.2017.09.023.

[35] J. Ahmed, S. Yildirim, M. Nowostaki, R. Ramachandra, O. Elezaj, and M. Abomohara, “GDPR compliant consent driven data protection in online social networks: a blockchain-based approach,” The 3rd International Conference on Information and Computer Technologies (ICICT 2020), San Jose, USA, 2020, ISBN: 978-1-7281-7283-5, doi:10.1109/ICICT50521.2020.00054.

[36] D. K. Tosh, S. Shetty, X. Liang, C. A. Kamhoua, K. A. Kwiat, and L. Njilla, “Security Implications of blockchain cloud with analysis of blockchain withholding attack,” The 17th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, Madrid, Spain, 2017, doi:10.1109/CCGRID.2017.111.

[37] D. Puthal, N. Malik, S. P. Mohanty, E. Kougianos, and C. Yang, “The blockchain as a decentralized security framework,” IEEE Consumer Electronics Magazine, vol. 7, no. 2, pp. 18-21, 2018, doi:10.1109/MCE.2017.2776459.

[38] D. Hofman, V. L. Lemieux, A. Joo, and D. A. Batista, “The margin between the edge of the world and infinite possibility: blockchain, GDPR and information governance,” Records Management Journal, vol. 29, no. 1/2, pp. 240257, 2019, doi:10.1108/RMJ-12-2018-0045.

[39] R. Teperdjian, “The puzzle of squaring blockchain with the General Data Protection Regulation,” Jurimetrics Journal, vol. 60, no. 3, 2020. [Online]. [Retrieved: March, 2021] https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3638736

[40] P. Hristov and W. Dimitrov, “The blockchain as a backbone of GDPR compliant frameworks,” The 8th International Multidisciplinary Symposium Conference on Challenges and opportunities for sustainable development through quality and innovation in engineering and research management (SIMPRO 2018), Petrosani, Romania, 2018. [Retrieved: March, 2021] https://www.researchgate.net/profile/Peyo-Hristov/publication/328576742_The_blockchain_as_a_backbone_of_GDPR_compliant_frameworks/links/5c27b3d6458515a4c700a92a/The-blockchain-as-a-backbone-of-GDPR-compliant-frameworks.pdf

[41] T. T. A. Dinh, R. Liu, M. Zhang, G. Chen, B. C. Ooi, and J. Wang, “Untangling blockchain: a data processing view of blockchain systems,” IEEE Transactions on Knowledge and Data Engineering, vol. 30, no. 7, pp. 1366-1385, 2018, doi:10.1109/TKDE.2017.2781227.

[42] Z. Xiong, Y. Zhang, D. Niyato, P. Wang, and Z. Han, "When mobile blockchain meets edge computing," IEEE Communications Magazine, vol. 56, no. 8, pp. 33-39, 2018, doi:10.1109/MCOM.2018.1701095.

[43] J. Y. Huumo, D. Ko, S. Choi, S. Park, and K. Smolander, "Where is current research on blockchain technology?: a systematic review," PLoS One, vol. 11, no. 10, 2016, doi:10.1371/journal.pone.0163477.

[44] H. Halpin and M. Piekarska, "Introduction to security and privacy on the blockchain," Paper presented at the IEEE European Symposium on Security and Privacy Workshops (EuroS&W), Paris, France, 2017, doi:10.1109/EuroSPW.2017.43.

[45] J. Moubarak, E. Filiol, and M. Chamoun, "On blockchain security and relevant attacks," The IEEE Middle East and North Africa Communications Conference (MENACOMM 2018), Jounieh, Lebanon, 2018, ISBN: 978-1-5386-1254-5, doi:10.1109/MENACOMM.2018.8371010.

[46] G. O. Karame, "On the security and scalability of Bitcoin's blockchain," The ACM/SIGSAC Conference on Computer and Communications Security (CCS 2016), Vienna, Austria, 2016, pp. 1861-1862, doi:10.1145/2976749.2976756.

[47] A. K. Fedorov, E. O. Kiktenko, and A. I. Lvovsky, "Quantum computers put blockchain security at risk," Nature, vol. 563, pp. 465-467, 2018, doi:10.1038/d41586-018-07449-z.

[48] R. Zhang and R. Xue, "Security and privacy on blockchain," ACM Computing Surveys, vol. 52, no. 3, 2019, doi:10.1145/3316481.

[49] V. J. Morkunas, J. Paschen, and E. Boon, "How blockchain technologies impact your business model," Business Horizons, vol. 62, no. 3, pp. 295-306, 2019, doi:10.1016/j.bushor.2019.01.009.

Analyzing Safety in Collaborative Cyber-Physical Systems: A Platooning Case Study

Manzoor Hussain, Nazakat Ali, Youngjae Kim, and Jang-Eui Hong
Department of Computer Science
Chungbuk National University
Cheongju, Republic of Korea
Email: {hussain, nazakatali, kyj}@selab.cbnu.ac.kr, jehong@chungbuk.ac.kr

Abstract— **Collaborative Cyber-Physical Systems (CCPS) are those systems in which several Cyber-Physical Systems (CPSs) collaborate to achieve a common goal. However, safety verification for collaborative CPSs is a significant challenge. The challenges occur due to unexpected operating conditions which are, by definition, unknown at development time or due to the lack of composite hazard analysis for collaborative CPSs. In this paper, we present an approach to perform safety analysis for collaborative CPSs by introducing an enhanced Fault Traceability and Propagation Graph based on composite hazard analysis. This graph enables to determine the fault source, propagation scope and required safety guards to mitigate the faults. We use the platooning system as a case study and modify the original VEhicular NeTwork Open Simulator (VENTOS) to verify safety for the platooning driving system in a variable environment (an unexpected event). Our simulation results show that after applying our defined safety guards, all the member vehicles in platoon managed to avoid the collision.**

Keywords-Matrix; Cyber-Physical Systems; Hazard analysis; Platooning System; Safety Verification.

I. INTRODUCTION

Cyber-Physical System (CPS) is a controlled, reliable, and extensible complex and connected physical system, in which the physical module of the system is integrated with computational, communicational, and control capabilities that can interact with the human through sensors [1].

The safety of multiple CPSs collaborating with other CPSs becomes a challenging task for safety engineers due to their complicated, diverse, variable, and uncertain operational environments. Therefore, a technique that may provide enough safety for collaborative CPSs operating in variable and uncertain environments is required. Despite ISO 26262 and IEC 61508 safety processes and procedures, the safety of multiple CPSs collaborating to achieve a common goal is a challenge as elaborated in [2]. Due to the variable and diverse operational environment of collaborative CPSs, safety assurance becomes a difficult task [3]. The unexpected behavior in collaborative CPSs can come from unintended behavior of the failure-free system due to its performance limitation or lack of robustness regarding the environmental variability (such as fog and rain) that may disturb the sensors and actuators or due to insufficient situational awareness. Collaborative CPSs, for example, platooning mostly operate in a variable, and uncertain environmental conditions such as extreme weather conditions in foggy and heavy raining scenarios.

The focus of our paper is to investigate the collaborative nature of CPSs, analyze safety issues emerging during the collaboration of CPSs due to variabilities, trace the faults originating from the system collaborating in CPSs, and analyze the impact of a fault on other systems in CPSs in detail.

In this paper, we enhance our previous Fault Traceability graph [11] by introducing new Fault Propagation and Traceability Graph (FPTG), Fault Propagation Graph (FPG), and Fault Back Traceability Graph (FBTG) to investigate the fault route, propagation scope of fault, fault origin and impact of fault other systems. This study is built on our previous work [11] that proposed a composite hazard analysis technique for collaborative CPSs based on the content relationships among the hazard analysis artifacts. We modified the original VENTOS [4] simulator to create hazardous scenarios such as fog, rain, and snow to validate our approach. After analyzing the hazards for platooning systems (an example of CCPSs) with FPTG, FPG, and FBTG, we verify the safe behavior of the platooning system at run-time by using the VENTOS simulator.

The remaining part of the paper is organized as follows: Section II presents the literature review. In Section III, we present the proposed approach, and Section IV concludes this paper with some future research directions.

II. RELATED WORK

Designing a CCPS is a thorny challenging work due to its highly integrated physical, information, and communication modules. It demands higher reliability and robustness than a common system. The authors in [5] proposed a conceptual framework called A2CPS (autonomous CPSs) aiming to design and implement an autonomous supervision and control system. The purpose of this proposed framework was to reduce the probability of vehicle collision with resilient safety measures in a run-time fashion and control loop process.

Medawar et al. [6] discussed the role of the run-time manager in SafeCOP to ensure continuous safety in truck platooning. The authors first specify the safety contracts based on the safety analysis of the local system, as well as the cooperative safety function. The study further argues that safety contracts must be examined during the design phase to check their validity. Zhang et. al [7] proposed a taxonomy that can be translated under the uncertainty of the predictive model. A self-healing model is proposed to ensure the sustainable safety of the CPSs. A domain-specific language (CyPhyML+) was proposed by [8] to identify the interaction component and their uncertainties in collaborative CPSs. This language is an extension of CyPhyML [9]. In this approach, the semantic unit for heterogenous component interaction is identified within the collaborative CPS. The primary objective of this approach was to present the safety component and identifying unknown component interaction in CPSs ensuring safety.

The behavior of a robot in a human-robot collaborative environment should be adaptable as per human actions as mentioned in [10]. The authors investigated the capability of the proposed architecture to ensure human safety in the production environment. The safety in human-robot collaboration is ensured through a closed-loop control system that is based on human vicinity to robots.

III. Proposed Approach

The collaborative nature of CPSs and their operations in dynamic and uncertain environments raise safety issues. Sustainable safety at run time in adverse weather conditions is a real safety concern. The hazard analysis in CPSs makes it possible for safety engineers to identify potential failures and provides safety guards to mitigate the faults in the system. Therefore, we propose an approach to analyze safe operability for collaborative CPSs as shown in Figure 1. In the first step of our approach, we analyze the behavior of collaborative CPSs and try to consider variability factors in the behavioral analysis of CPSs at development time. In collaborative CPSs, failure in one CPS may affect other CPSs with whom it collaborates.

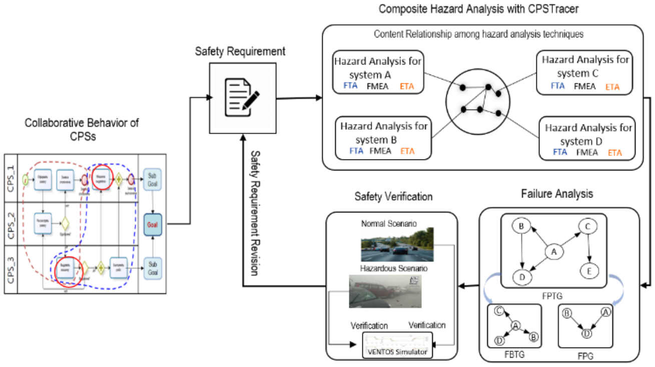

Composite Hazard Analysis with CPSTracer
Content Relationship among hazard analysis techniques
Safety Requirement
Hazard Analysis for system A FTA FMEA ETA
Hazard Analysis for system C FTA FMEA ETA
Hazard Analysis for system B FTA FMEA ETA
Hazard Analysis for system D FTA FMEA ETA
Collaborative Behavior of CPSs
CPS_1
CPS_2
CPS_3
Sub Goal
Goal
Sub Goal
Safety Requirement Revision
Safety Verification
Failure Analysis
Normal Scenario
Hazardous Scenario
Verification
Verification
VENTOS Simulator
B
C
A
D
E
FPTG
C
A
D
B
FBTG
B
A
D
FPG

Figure 1. The proposed approach for analyzing safety in CCPS.

We introduced content-based relationships in our previous work [11] among the hazard analysis technique to envision the relationship among faults coming from different systems and the impact of a specific fault on other systems collaborating in CPSs. A single hazard analysis technique is not sufficient to ensure the safety of collaborative CPSs. Composite hazard analysis is necessary to prevent such failures by introducing safety guards in time. Therefore, to perform safety analysis of collaborative CPSs based on composite hazard analysis technique, we introduce conceptual Fault Traceability Graphs (FTG) in our previous work [11] to visualize the relationship between the faults and safety guards. However, this approach does not consider the variabilities such as environmental (fog, rain, and snow, etc.), temporal, infrastructural, and spatial variabilities. Also, the graph does not provide information about the source of the faults, and the information about the hazard analysis through which the fault is analyzed.

In this paper, we extend the FTG and developed FPTG. The FPTG aims to reflect the variabilities in CPSs and to visualize the impact of specific faults on other systems in CCPS, propagation scope, and origin of the faults. The FPTG shows the impact of a failure on other functionalities of collaborative systems and it shows the backward traceability of a fault as well, which is called FBTG. Another graph called FPG is also proposed to show all possible impacts of a specific fault on other systems. In the following subsections, we explain our proposed approach in detail.

A. Collaborative Behaviors of CPSs

To analyze the collaborative behavior of CPSs, we take the platooning CPS as a running example. In the platooning system, several vehicles form a platoon where the follower vehicle of the platoon maintains a short inter-vehicle distance with the preceding vehicle to improve traffic flow, reduce traffic congestion, and reduce fuel consumption [12]. The platooning system uses Cooperative Adaptive Cruise Control (CACC) where platooning vehicles communicate with each other to create synergy in their cooperation. The vehicles in the platooning system can also use an Adaptive Cruise Control (ACC) unit when necessary. In ACC mode, the platooning vehicles rely on onboard sensors instead of depending on other vehicles. As the distance among the vehicles is very short, therefore, the leader’s failure can be propagated to other vehicles, as a result, a hazardous scenario may occur.

B. Safety Requirements

The safety requirements are those requirements that are defined to reduce the risk in any system. These requirements are also like other requirements, first specified at a high level, for example, it is needed to reduce a given risk. These requirements must be refined and then supplied to the designer. In our approach, we first analyzed the collaborative nature of CPSs, then, we extracted the safety requirements to reduce the identified faults and ensure an acceptable level of safety in collaborative CPSs. Each safety requirement is then supplied to composite hazard analysis as an input. Then, we analyze the collaborative CPSs with our composite hazard analysis tool to identify the potential faults based on the safety requirements. After performing the composite hazard analysis, we perform the failure analysis and verified whether the identified faults are removed from the system or not. This process is a loop process and this process is continued until an acceptable level of safety is achieved and the safety requirements are also revised according to fault status in the collaborative CPSs.

C. Case Study: Composite Hazard Analysis of Platooning with CPSTracer

In the platooning CPS, where the movement of vehicle group collaborates to reduce the inter-vehicle distance which benefits the better usage of road infrastructure by allowing more vehicles to use a given stretch of road, improve energy efficiency by reducing the aerodynamic drag [13]. On the other hand, reducing the inter-vehicle distance also leads to creating safety concerns in vehicles participating in the platooning. The safety of collaborative CPSs can be ensured by analyzing the safety of the system considering the potential uncertainties. The main objective of hazard analysis is to identify the potential hazards, analyze the faults, and measurement of possible damage. As mentioned, a composite hazard analysis technique can trace fault propagation in collaborative CPSs. In our previous work [11], we defined four relationships (i.e., *influence relationship, inheritance, overlap, and supplement relationship*) among Fault Tree Analysis (FTA), Failure Mode and Effect Analysis (FMEA), and Event Tree Analysis (ETA). The definition of relationships are as follows

Influence Relationship: This relationship exists among the faults of the participating system in collaborative CPSs in which a fault of a system causes the failure of another participating system.

Inheritance Relationship: This relationship exists when two or more system participating in collaborative CPSs shares the same operational and functional constraints. This relationship also exists among the faults of the participation system in CCPSs.

Overlap Relationship: This relationship exists among the faults and outcomes/consequences of failure in collaborative CPSs. There exist overlap relationships when the consequences of the failure of a system are the same as the consequences of the failure of another system.

Supplement Relationship: This relationship exists among the safety guards and failures of the system in collaborative CPSs. When a system has safety guards to cope with the failure of another system in collaborative CPSs, this relationship is then established. This means that the safety guard for the failure of a system can be supplied to another identical failure of the system in CPSs.

We developed a composite hazard analysis tool (i.e., CPSTracer) to analyze the potential hazards for collaborative CPSs. This tool helps to analyze the potential hazard with variability that a CCPS may face often. In our previous work [14], we extended FTA, FMEA, and ETA a.k.a. v_FTA,v_FMEA, and v_ETA to capture the variability in collaborative CPSs. Therefore we used our extended FTA, FMEA, and ETA to analyze the potential hazards due to variabilities (e.g., *environmental, infrastructural, temporal, and spatial variability*) for the platooning case study.

FTA is widely used for hazard and risk assessment in CPS. The FMEA is a structured method for system safety analysis to identify, evaluate, and score the potential failure for the system and its effects. ETA shows all possible outcomes stemming from a mishap event and takes into account additional events and factors i.e., whether or not installed safety barriers are working. ETA can be used to identify possible potential accident scenarios and sequences in a complex system. In the first step of the composite hazard analysis technique, an FTA is performed to identify the root cause of the failure of the platooning system. Let us consider that, one of the reasons for platooning failure is *Car Collision* (i.e., a top event in FTA). The top event in FTA is a failure of the system as a whole, which is in the case of the platooning, the participant vehicles were not collaborating, and as a result *Car Collision* has happened. An FTA consisting of five levels for the platooning is shown in Figure 2. The intermediate events and basic events are the root cause of the top event in FTA (i.e., *Car Collision*). Let us take the example of an intermediate event (i.e., *Collision event*) to analyze its root

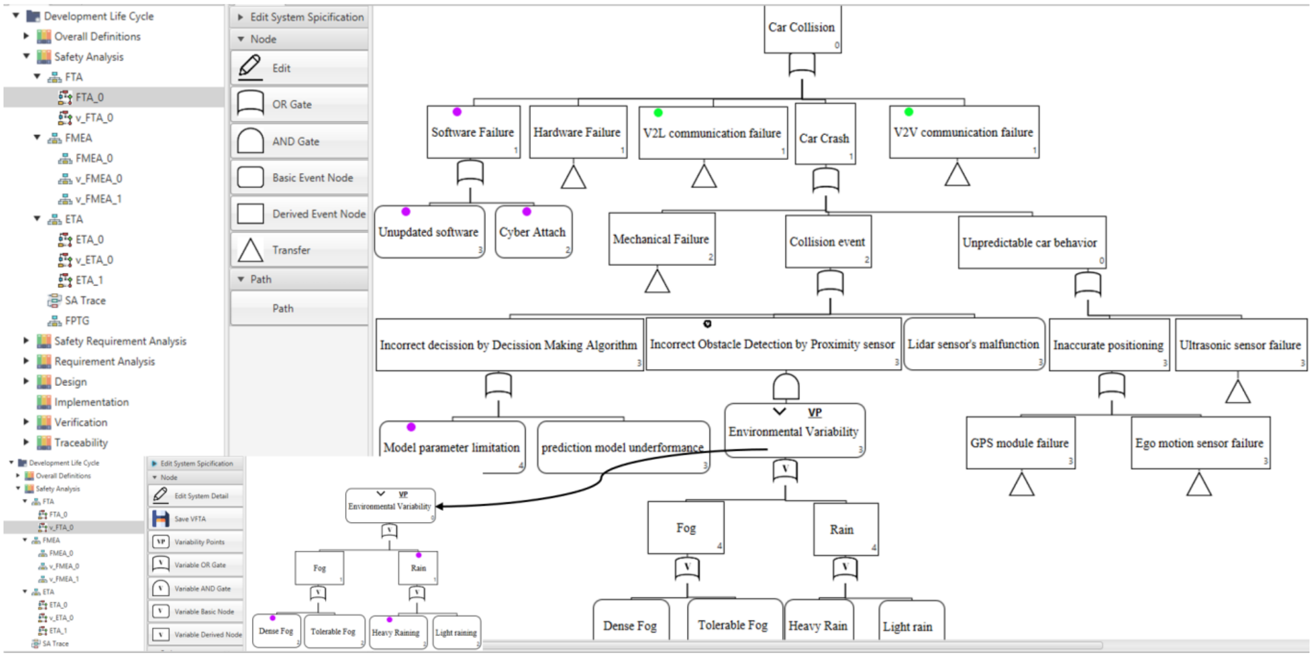
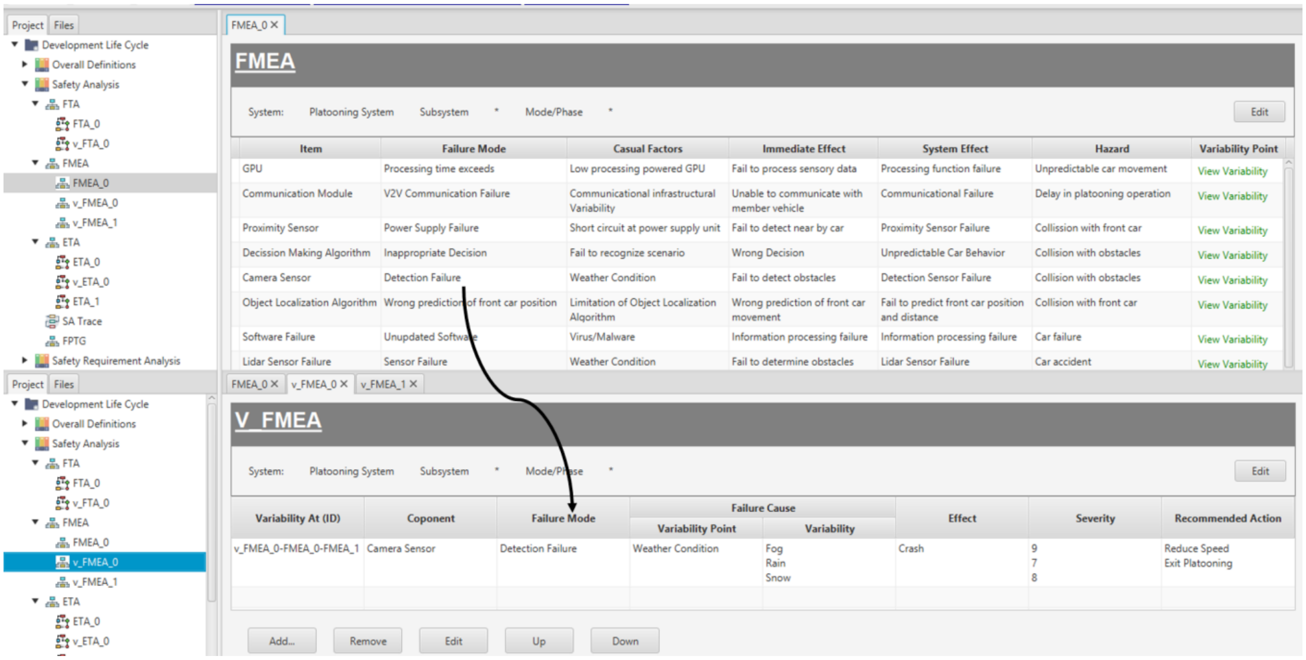


Figure 2. Hazard analysis of the platooning system with FTA and v_FTA.



FMEA

System: Platooning System Subsystem * Mode/Phase * Edit

Item	Failure Mode	Casual Factors	Immediate Effect	System Effect	Hazard	Variability Point
GPU	Processing time exceeds	Low processing powered GPU	Fail to process sensory data	Processing function failure	Unpredictable car movement	View Variability
Communication Module	V2V Communication Failure	Communicational infrastructural Variability	Unable to communicate with member vehicle	Communicational Failure	Delay in platooning operation	View Variability
Proximity Sensor	Power Supply Failure	Short circuit at power supply unit	Fail to detect near by car	Proximity Sensor Failure	Collission with front car	View Variability
Decission Making Algorithm	Inappropriate Decision	Fail to recognize scenario	Wrong Decision	Unpredictable Car Behavior	Collision with obstacles	View Variability
Camera Sensor	Detection Failure	Weather Condition	Fail to detect obstacles	Detection Sensor Failure	Collision with obstacles	View Variability
Object Localization Algorithm	Wrong prediction of front car position	Limitation of Object Localization Algorithm	Wrong prediction of front car movement	Fail to predict front car position and distance	Collision with front car	View Variability
Software Failure	Unupdated Software	Virus/Malware	Information processing failure	Information processing failure	Car failure	View Variability
Lidar Sensor Failure	Sensor Failure	Weather Condition	Fail to determine obstacles	Lidar Sensor Failure	Car accident	View Variability

V_FMEA

System: Platooning System Subsystem * Mode/Phase * Edit

Variability At (ID)	Coponent	Failure Mode	Failure Cause		Effect	Severity	Recommended Action
			Variability Point	Variability			
v_FMEA_0-FMEA_0-FMEA_1	Camera Sensor	Detection Failure	Weather Condition	Fog Rain Snow	Crash	9 7 8	Reduce Speed Exit Platooning

Figure 3. Hazard analysis of platooning system with FMEA and v_FMEA..

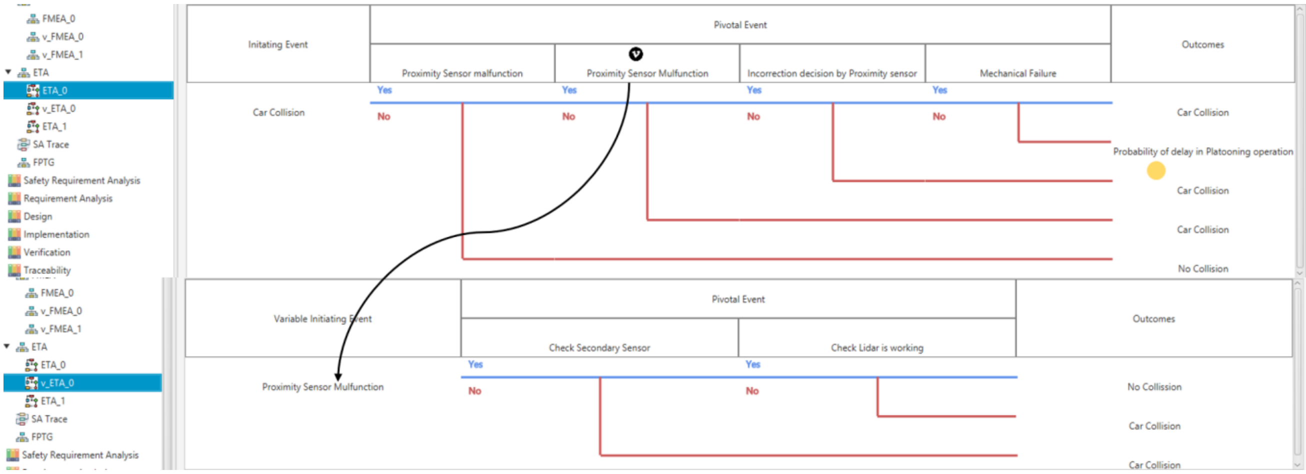


Figure 4. Hazar analysis of platooning with ETA and v_ETA.

cause in detail. We assume that the root cause of the *Collision vent* may be the *Lidar sensor's malfunction* or *incorrect decision by decision making algorithm* or may be due to *Incorrect Obstacle Detection by Proximity Sensor*. In general, FTA doesn't consider variability. The traditional FTA cannot capture the variability factors that lead to unexpected events at run time. We need to consider the variabilities while performing hazard analysis for collaborative CPSs. In our case study, we further investigated the intermediate event *Incorrect Obstacle Detection by Proximity Sensor* to find the more basic reason due to which *Incorrect Obstacle Detection by Proximity Sensor* event has happened with our extended FTA a.k.a v_FTA. Hence we come up with more basic events like *Dense Fog*, *Tolerable Fog*, *Heavy Rain*, and *Light Raining*.

In the second step of our composite hazard analysis technique, we analyzed the potential hazards for the platooning with FMEA and extended FMEA also known as v_FMEA. We also introduce a new column in FMEA. This column contains the safety guards for each fault. Let us take the example of the *Camera Sensor* failure, the causal factors of the *Camera Sensor Failure* may be due to *weather conditions*. To explore the more basic cause of *Camera Sensor Failure*, we investigate the more basic cause of *Camera Sensor Failure* with the v_FMEA. As we can see in Figure 3, after analyzing the *Camera Sensor Failure* with the v_FMEA, it is clear that the more reasonable causes of *Camera Sensor failure* are due to *Fog, Rain, and snow*.

In the last step, we analyze the platooning with ETA in our composite hazard analysis tool. We analyze the variability factor for *Proximity Sensor Malfunction* with v_ETA. We investigated the *Proximity Sensor Malfunction* for variability. Figure 4 shows the hazard analysis of platooning with ETA, as well as v_ETA.

D. Failure Analysis

Collaborative CPSs require more effective safety analysis to provides better fault traceability, fault propagation, fault sources, impact analysis of the fault, and potential safety guards for faults. The identification of fault propagation is a challenging task especially in collaborative CPSs for safety engineers. The proposed FPTG can be used as a means of failure analysis in collaborative CPSs because it can visualize the potential faults that may lead to the failure of collaborative CPSs. We developed an algorithm that detects the content relationship among the hazard analysis artifacts and generates the FPTG.

The FTPG is a directed graph in which the vertices represent the faults and safety guards, and the edges denote the relationships among the faults also relationships among faults and safety guards. Each node on the FPTG has complete information about the fault, its origin, and the hazard analysis technique used to analyze the faults. The colored edges on FPTG show the four content relationships as mentioned earlier. The arrow direction on FPTG determines the propagation of faults in collaborative CPSs.

As CCPS consists of highly interconnected systems, a fault in a participant system may lead to activating many other faults in other systems. The information on the nodes of FPTG can also help the safety engineers to determine where exactly a safety guard should be provided to eliminate the fault and stop its propagation to another system. The fault traceability determines the fault routes in collaborative CPSs. It is necessary to demonstrate that a safety-critical system must fulfill the safety goal, and all identified potential hazards were eliminated. The FPTG can identify the safety guards to mitigate potential faults. Both FPG and FBTG are also directed graphs. In our developed tool, after generating the FPTG we can select any fault on FPTG to know about its propagation scope and its route by clicking on a particular fault. A separate subgraph also known as FPG is generated for that specific fault which tells us the propagation route of that specific fault. It also clearly depicts how much a certain fault on FPGT is critical for the collaborative system's safety. The FBTG shows the traceability of a specific fault. By clicking on any fault on FPG, we can generate FBTG which shows the back traceability of that specific fault.

The relationships on FPTG, FPG, and FBTG are illustrated by color legends, as shown in Figure 5, Figure 6, and Figure 7. The *inheritance relationship* is represented with a green-colored edge, *influence relationship* with a purple edge, *overlap relationship* with a yellow edge, and *supplement relationship* with a red-colored edge. All variability nodes like environmental variability in the platooning system are represented by a black-bordered white colored circle to reflect the variability on FPTG, FPG, and FBTG. The node *Dense Fog.[Platooning System.v_FTA_0]* is an example of variability in Figure 5. In our case study, the node *Wrong Decision.[Platooning System.FMEA_0]*, influences the node *Collision event.[Platooning System.FTA_0]* and *Unpredictable car behavior.[Platooning System.FTA_0]*. Same as the node *V2V communication failure.[Platooning System.FTA_0] and V2L communication failure.[Platooning System.FTA_0]* inherits *Communicational Failure.[Platooning System.FMEA_0]*. As discussed earlier, the Overlap relationship exists when the failures of the

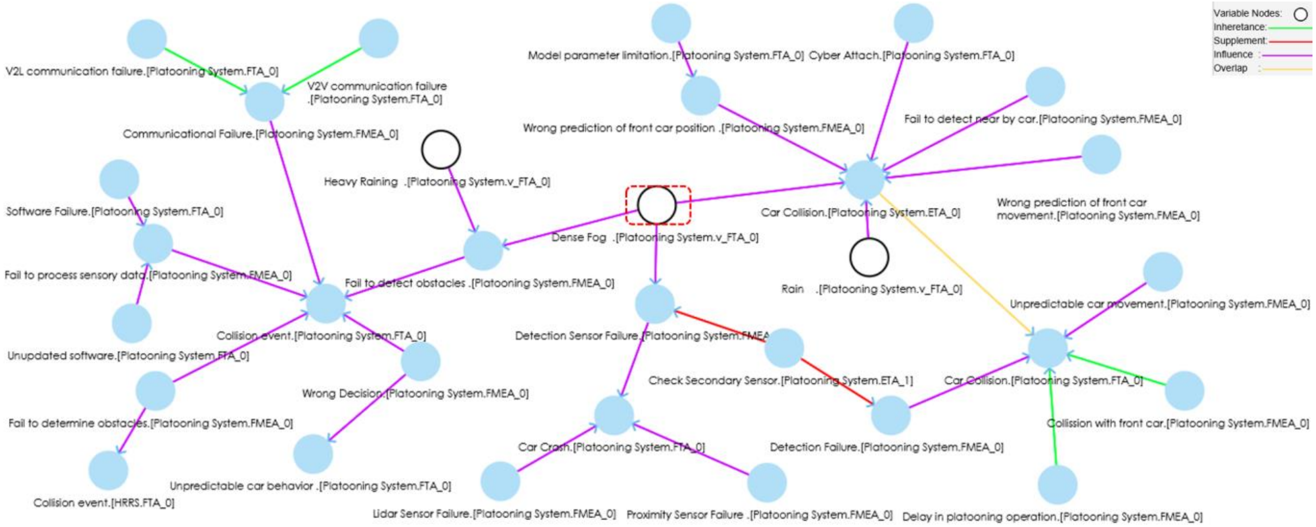

V2L communication failure.[Platooning System.FTA_0]
V2V communication failure
.[Platooning System.FTA_0]
Communicational Failure.[Platooning System.FMEA_0]
Heavy Raining .[Platooning System.v_FTA_0]
Software Failure.[Platooning System.FTA_0]
Fail to process sensory data.[Platooning System.FMEA_0]
Fail to detect obstacles .[Platooning System.FMEA_0]
Collision event.[Platooning System.FTA_0]
Unupdated software.[Platooning System.FTA_0]
Wrong Decision.[Platooning System.FMEA_0]
Fail to determine obstacles.[Platooning System.FMEA_0]
Unpredictable car behavior .[Platooning System.FTA_0]
Collision event.[HRRS.FTA_0]
Model parameter limitation.[Platooning System.FTA_0]
Cyber Attach.[Platooning System.FTA_0]
Wrong prediction of front car position .[Platooning System.FMEA_0]
Fail to detect near by car.[Platooning System.FMEA_0]
Dense Fog .[Platooning System.v_FTA_0]
Car Collision.[Platooning System.ETA_0]
Wrong prediction of front car
movement.[Platooning System.FMEA_0]
Rain .[Platooning System.v_FTA_0]
Unpredictable car movement.[Platooning System.FMEA_0]
Detection Sensor Failure.[Platooning System.FMEA
Check Secondary Sensor.[Platooning System.ETA_1]
Car Collision.[Platooning System.FTA_0]
Collission with front car.[Platooning System.FMEA_0]
Car Crash.[Platooning System.FTA_0]
Detection Failure.[Platooning System.FMEA_0]
Lidar Sensor Failure.[Platooning System.FMEA_0]
Proximity Sensor Failure .[Platooning System.FMEA_0]
Delay in platooning operation.[Platooning System.FMEA_0]
Variable Nodes: ○
Inheretance:
Supplement:
Influence :
Overlap :

Figure 5. Fault Propagation and Traceability Graph for platooning.

systems in CPSs are the same. The node *Car Collision.[Platooning System.FTA_0]* overlaps the node *Car collision.[Platooning System.ETA_0]*. The Supplement Relationship provides safety guards. The node *Detection Failure.[Platooning System.FMEA_0]* is supplemented by *Check Secondary Sensor.[Platooning System.ETA_1]*. This means that the *Check Secondary Sensor.[Platooning System.ETA_1]* is supplied as safety guards to mitigate the effect of *Detection Failure.[Platooning System.FMEA_0]* and so on. Figure 5 shows the FPTG for platooning.

The information within the square brackets represents the source of faults and the hazard analysis technique used to analyze the system to perform hazard analysis. For example in the node *Collision event [Platooning system.FTA_0]* on FTPG, *Collision event* is the description of the fault, *Platooning system* within the square bracket represents the system being analyzed and the origin of the fault. *FTA_0* represents, the Fault Tree Analysis technique used to analyze the platooning system.

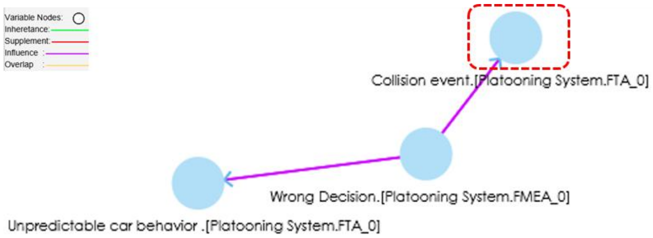

Variable Nodes: ○
Inheretance:
Supplement:
Influence :
Overlap :
Collision event.[Platooning System.FTA_0]
Wrong Decision.[Platooning System.FMEA_0]
Unpredictable car behavior .[Platooning System.FTA_0]

Figure 6. Fault Propagation Graph of platooning.

By clicking on a particular fault on FPTG, we can generate FPG. The algorithm generates the FPG which represents all possible impacts of a fault on other systems. This helps the safety engineers to make possible steps to mitigate the faults by apply suitable safety guards. From FPTG, we clicked the node *Wrong Decision.[Platooning System.FMEA_0*] to generate the FPG, as shown in Figure 6.

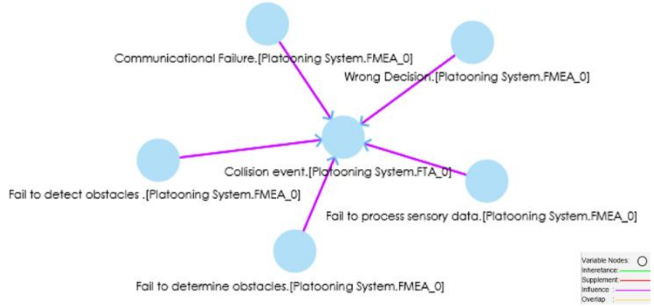

Communicational Failure.[Platooning System.FMEA_0]
Wrong Decision.[Platooning System.FMEA_0]
Collision event.[Platooning System.FTA_0]
Fail to detect obstacles .[Platooning System.FMEA_0]
Fail to process sensory data.[Platooning System.FMEA_0]
Fail to determine obstacles.[Platooning System.FMEA_0]
Variable Nodes: ○
Inheretance:
Supplement:
Influence :
Overlap :

Figure 7. Fault Back Traceability Graph of platooning.

To know about the root cause of the occurrence of a specific fault, we just need to click on the nodes on FPG. From FPG we clicked on the node *Collision event.[Platooning System.FTA_0]* to generate the FBTG. Figure 7 shows the FBTG for the node *Collision event.[Platooning System.FTA_0].*

E. *Safety Verification*

In platooning, vehicles may face several environmental variabilities such as fog, rain, snow, and rushing objects on the road that might affect the vision of platooning vehicles, and collision may occur. The effect of environmental variability on the platooning vehicle's vision may lead to the collision of the whole platooning system. For example, if the platooning leader's vision sensor is affected due to dense fog or heavy rain then it may cause the collision of follower vehicles because the distance between platooning vehicles is supposed to be short.

After performing the hazard analysis for platooning with our developed tool we verify the behavior of platooning. The safety verification is necessary to confirm whether or not the identified faults in the system were removed. During the hazard analysis of platooning, we found that environmental variabilities such as fog, rain, and snow affects the vision (sensors) of cars in the platooning. We identified the potential faults that lead to the platooning collision during our composite hazard analysis. We first present a normal scenario, a hazardous scenario, safe scenario by applying a defined safety guard and then simulate these scenarios in VENTOS. In our simulation, we implement a platoon of size 5 (one leader denoted V0 and four followers denoted by V1..V4).

Normal Scenario: Five vehicles are running in a platoon on a highway with a speed of 25km/h (max speed in VENTOS simulator), inter-vehicles distance (minimum) 4m, and V2V (vehicle-to-vehicle) and L2V (leader to vehicles) communication modes. The leader communicates with the roadside unit and obtains road status information and receives no accident or traffic congestion information. The platoon continues to drive on its route under normal weather conditions. The speed and inter-vehicle for the normal scenario of the platooning system are shown in Figure 8.

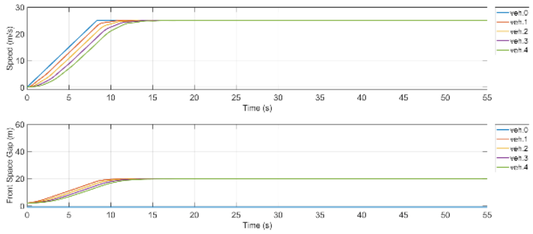


Figure 8. Speed and inter-vehicle distance for the normal scenario.

Hazardous Scenario: The vehicles in the platoon were on the way to their final destinations under normal weather conditions. We modify the original VENTOS simulator to create unexpected scenarios such as fog, rain, and snow. At some point, the platoon faces dense fog, and the platoon leader transmitted a reduction of speed command to its followers. The platoon reduced its speed accordingly. Suddenly, the platoon leader collided with a non-platooning vehicle due to its perception failure. The immediate follower of the leader also collided with the leader while the last three platooning vehicles managed to stop without collision. The vehicles changed their mode from CACC to ACC, changed their lane, and continue to drive. Figure 9 shows the simulation result of a hazardous scenario in terms of speed and inter-vehicle space.

As we see, at the time point 25, the leader vehicle faced dense fog and reduced speed gradually. At time point 27, a non-platooning vehicle suddenly came in front of the leader vehicle and a collision has happened due to the inaccurate decision of the proximity sensor. However, vehicles V2, V3, and V4 managed to stop without collision and changed their mode to ACC, changed their lane, and formed a new platoon to continue their journey.

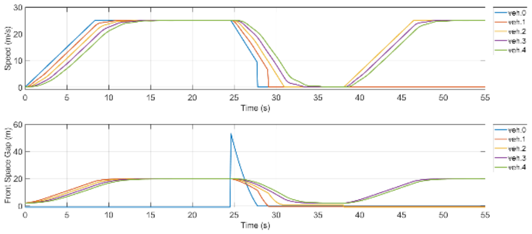


Figure 9. Speed and inter-vehicle distance for the hazardous scenario.

Safe Scenario: The vehicles in the platoon were on the way to their final destinations under normal weather conditions. At some point, the platoon faces dense fog, and the platoon leader transmitted a reduction of speed command to its followers. The followers reduced their speed as directed. The vehicles in the platoon were moving under fog by reducing their speed, a non-platooning vehicle suddenly changed its lane and came in front of the platoon.

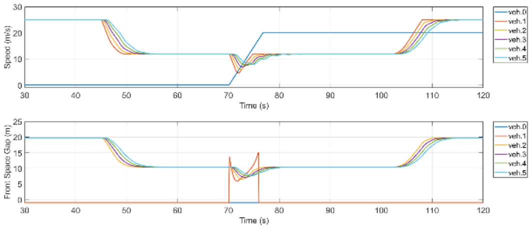


Figure 10. Speed and inter-vehicle distance for safe scenario

The leader vehicles detected it under dense fog and reduced its speed further to avoid the collision, by applying a safety guard, i.e., ‘Urgent Brake”. Figure 10 shows the implemented safe scenario in VENTOS.

IV. Conclusion

Collaborative CPSs are systems where multiple CPSs collaborate to achieve a common goal. However, safety remained a thorny challenge in collaborative CPSs due to the complex, diverse and variable operational environment of CCPS. The failure in one CPS of a collaborative CPSs may lead to the failure of other participant systems. Therefore, we proposed FPTG, FPG, and FBTG based on composite hazard analysis and content-based relationship to perform safety analysis. It enables to determine the fault route, the origin of faults, and its impact on other systems in a CCPS. We perform the safety analysis of platooning systems considering variability by using our developed tool and took the advantage of the VENTOS to verify the safe behavior of a platooning system. We are working on a learning-based approach to ensure safety verification in an on-the-fly situation by predicting the potential misbehavior in CPSs.

Acknowledgment

This work is supported by the Korea National Research Foundation of the Ministry of ICT and Science (NRF-2020R1A2C1007571).

References

[1] N. Ali and J. E. Hong, “Failure detection and prevention for cyber-physical systems using ontology-based knowledge base,” Computers, vol. 7, no. 4, Dec. 2018, p.68, doi: 10.3390/computers7040068.

[2] X. Lyu, Y. Ding, and S. H. Yang, “Safety and security risk assessment in cyber-physical systems,” IET Cyber-Physical Systems: Theory and Applications. 2019, pp.221-232, doi: 10.1049/iet-cps.2018.5068.

[3] F. Platbrood and O. Gornemann, “Safe robotics-safety in collaborative robot systems,” Sick AG, Waldkirch, Ger., vol. 980, 2017.

[4] M. Amoozadeh, H. Deng, C. N. Chuah, H. M. Zhang, and D. Ghosal, “Platoon management with cooperative adaptive cruise control enabled by VANET,” *Veh. Commun.*, 2015, pp.110-123, doi: 10.1016/j.vehcom.2015.03.004.

[5] J. K. Naufal et al., “A2CPS: A Vehicle-Centric Safety Conceptual Framework for Autonomous Transport Systems,” IEEE Trans. Intell. Transp. Syst., 2018, pp.1925-1939, doi: 10.1109/TITS.2017.2745678.

[6] S. Medawar, D. Scholle, and I. Šljivo, “Cooperative safety-critical CPS platooning in SafeCOP,” 2017, pp. 1-5, doi: 10.1109/MECO.2017.7977210.

[7] M. Zhang, B. Selic, S. Ali, T. Yue, O. Okariz, and R. Norgren, “Understanding uncertainty in cyber-physical systems: A conceptual model,” 2016, pp. 247-264, doi: 10.1007/978-3-319-42061-5_16.

[8] H. Daneth, N. Ali, and J. E. Hong, “Automatic Identifying Interaction Components in Collaborative Cyber-Physical Systems,” 2019, pp. 197-203, doi: 10.1109/APSEC48747.2019.00035.

[9] G. Simko, D. Lindecker, T. Levendovszky, S. Neema, and J. Sztipanovits, “Specification of cyber-physical components with formal semantics - Integration and composition,” 2013, pp. 471-487, doi: 10.1007/978-3-642-41533-3_29.

[10] N. Nikolakis, V. Maratos, and S. Makris, “A cyber-physical system (CPS) approach for safe human-robot collaboration in a shared workplace,” *Robot. Comput. Integr. Manuf.*, 2019, pp.233-243, doi: 10.1016/j.rcim.2018.10.003.

[11] H. Daneth, N. Ali, and J. E. Hong, “Towards enhancement of fault traceability among multiple hazard analyses in cyber-physical systems,” Proc. IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC), IEEE Press, July 2019, pp. 458-464, doi: 10.1109/COMPSAC.2019.10249.

[12] D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, “A survey on platoon-based vehicular cyber-physical systems,” *IEEE Commun. Surv. Tutorials*, 2016, pp.263-284, doi: 10.1109/COMST.2015.2410831.

[13] J. Axelsson, “Safety in vehicle platooning: A systematic literature review,” *IEEE Trans. Intell. Transp. Syst.*, 2017, pp.1033-1045, doi: 10.1109/TITS.2016.2598873.

[14] N. Ali, M. Hussain, and J.-E. Hong, “Analyzing Safety of Collaborative Cyber-Physical Systems Considering Variability,” *IEEE Access*, 2020, pp.162701-162713, doi: 10.1109/access.2020.3021460.

On the Composability of Behavior Driven Acceptance Tests

Tugkan Tuglular
Department of Computer Engineering
Izmir Institute of Technology
Izmir, Turkey
mail: tugkantuglular@iyte.edu.tr

***Abstract*—This paper proposes a model-based approach for composition of Behavior Driven Acceptance Tests (BDATs) using Event Sequence Graphs (ESGs). ESGs are used to generate test sequences automatically. For the composition process of BDATs, the ESG formalism is extended with tags and the technique called elimination of tags by combination is introduced for tagged ESGs. The proposed approach improves testability of existing behavior driven acceptance test suites. It is validated through a real-life example. The results demonstrate the feasibility of the proposed approach.**

Keywords-model-based testing; event sequence graphs; behavior driven acceptance tests; Gherkin.

I. Introduction

Behavior Driven Development (BDD) is focused on defining fine-grained specifications of the behavior of the targeted system [1]. In BDD, tests are clearly written using a specific ubiquitous language, such as Gherkin [2]. For developing Behavior Driven Acceptance Tests (BDATs), there are environments like Cucumber [2], which forces testers to use a test template using Gherkin language and environments like Gauge [3], which does not impose any language. The scope of this study is BDATs developed in Gherkin.

Although Gherkin and its scenario template helps test designers in writing test cases, they do not guide test designers in test objectives. The test designer either develops BDATs in an ad-hoc manner or follows rules of thumb such as happy path testing and negative testing. In either case, the test designer is not certain about the completeness or coverage of the BDAT test suite. As a solution, this paper proposes to transform Gherkin scenarios into formal test models, so that the test designer can work on completeness and coverage of BDATs.

The proposed approach assumes that clauses written in Gherkin can be represented by events. In that case, an event-based formal model would fit better to BDATs. Therefore, this paper proposes the use of Event Sequence Graphs (ESGs) for modeling BDATs. To model a BDAT as an ESG, ESGs are extended with tags. This is one of the novelties presented in this paper. Another novelty presented here is the process of finding missing BDATs. To find missing BDATs, the proposed approach follows elimination of tags by combination. After the missing BDATs are completed, an ESG without any tags is obtained. The proposed approach is explained with a running example in Section III. For evaluation, a BDAT test suite is selected from Github^{TM} and the proposed approach is applied to this test suite. The results are shared in Section IV.

The paper is organized as follows: In the next section, the formal definitions of ESGs are given along with examples and figures. The proposed approach is explained in Section III. Section IV gives an evaluation of the proposed approach along with a discussion in Section V. Section VI outlines related work, and the last section concludes the paper.

II. Fundamentals

A. Gherkin

Gherkin uses a set of special keywords to give structure and meaning to executable specifications [2]. It provides the behavior definitions of the intended software not only to product owners and business analysts, but also to developers and testers [4]. Gherkin is a line-oriented language in terms of structure and each line has to be divided by the Gherkin keyword except feature and scenario descriptions [2]. In this paper, some of the Gherkin keywords; namely *Feature*, *Scenario*, *Given*, *When*, *And*, *Then*, are utilized. Throughout the paper, the terms Gherkin scenario, scenario, and BDAT are used interchangeably.

Tests should be independent of each other so that they can be run in any order or even in parallel. This principle is also applied in developing BDATs. So, each BDAT should be run manually or automatically independent of other BDATs. However, they should also be composable so that it will be possible to execute a BDAT after a related one.

B. Event Sequence Graphs

A model of the system, which requires the understanding of its abstraction, helps in testing its behavior. A formal specification approach that distinguishes between legal and illegal situations is necessary for acceptance testing. These requirements are satisfied by event sequence graphs [5].

Differing from the notion of finite-state automata, inputs and states are merged in ESG, hence they are turned into "events" to facilitate the understanding and checking the external behavior of the system. Thus, vertices of the ESG represent events as externally observable phenomena, e.g., a user action or a system response. Directed edges connecting two events define allowed sequences among these events [5]. Definitions from 1 to 3 and related examples and explanations along with Figure 1 are taken exactly as they are from [6]-[9].

Definition 1. An *event sequence graph ESG = (V, E, Ξ, Γ)* is a directed graph where $V \neq \emptyset$ is a finite set of vertices (nodes), $E \subseteq V\times V$ is a finite set of arcs (edges), $\Xi,\Gamma \subseteq V$ are finite sets of distinguished vertices with $\xi \in \Xi$, and $\gamma \in \Gamma$, called entry nodes and exit nodes, respectively, wherein $\forall v \in V$ there is at least one sequence of vertices $\langle \xi, v_0, \ldots, v_k\rangle$ from each $\xi \in \Xi$ to $v_k = v$ and one sequence of vertices $\langle v_0, \ldots, v_k, \gamma\rangle$ from $v_0 = v$ to each $\gamma \in \Gamma$ with $(v_i, v_{i+1}) \in E$, for $i = 0, \ldots, k\text{-}1$ and $v \neq \xi, \gamma$.

To mark the entry and exit of an ESG, all $\xi \in \Xi$ are preceded by a pseudo vertex '[' $\notin V$ and all $\gamma \in \Gamma$ are followed by another pseudo vertex ']' $\notin V$. The semantics of an ESG are as follows. Any $v \in V$ represents an event. For two events $v, v' \in V$, the event v' must be enabled after the execution of v iff $(v, v') \in E$. The operations on identifiable components of the GUI are controlled and/or perceived by input/output devices, i.e., elements of windows, buttons, lists, checkboxes, etc. Thus, an event can be a user input or a system response; both of them are elements of V and lead interactively to a succession of user inputs and expected desirable system outputs.

Example 1. For the ESG given in Figure 1: $V=\{a,b,c\}$, $\Xi=\{a\}$, $\Gamma=\{b\}$, and $E = \{(a,b), (a,c),(b,c),(c,b))\}$. Note that arcs from pseudo vertex [and to pseudo vertex] are not included in E.

Furthermore, α(initial) and ω(end) are functions to determine the initial vertex and end vertex of an ES, e.g., for ES= $(v_0, \ldots, v_k)$ initial vertex and end vertex are $\alpha(\text{ES})=v_0$, $\omega(\text{ES})=v_k$, respectively. For a vertex $v\in V$, $N^+(v)$ denotes the set of all *successors* of v, and $N^-(v)$ denotes the set of all *predecessors* of v. Note that $N^-(v)$ is empty for an entry $\xi\in\Xi$ and $N^+(v)$ is empty for an exit $\gamma\in\Gamma$.

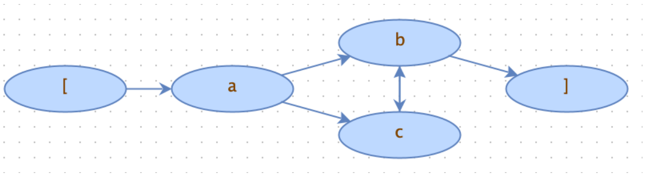


Figure 1. An ESG with a as entry and b as exit and pseudo vertices [,].

Definition 2. Let V, E be defined as in Definition 1. Then, any sequence of vertices $\langle v_0, \ldots, v_k\rangle$ is called an *event sequence* (*ES*) iff $(v_i, v_{i+1}) \in E$, for $i=0, \ldots, k\text{-}1$.

The function *l*(*length*) of an ES determines the number of its vertices. In particular, if $l(\text{ES})=1$ then $\text{ES}=(v_i)$ is an ES of length 1. Note that the pseudo vertices [and] are not considered in generating any ESs. Neither are they included in ESs nor considered to determine the initial vertex, end vertex, and length of the ESs. An ES = $\langle v_i, v_k\rangle$ of length 2 is called an *event pair* (EP).

Definition 3. An *ES* is a *complete ES* (or, it is called a *complete event sequence*, *CES*), if $\alpha(\text{ES})=\xi\in\Xi$ is an entry and $\omega(\text{ES})=\gamma\in\Gamma$ is an exit.

A CES may or may not invoke no interim system responses during user-system interaction. If it does not, that means that it consists of consecutive user inputs and only a final system response. CESs represent walks from the entry of the ESG to its exit, realized by the form (initial) user inputs → (interim) system responses → ⋯ (interim) user inputs → (interim) system responses → ⋯ → (final) system response.

III. Proposed Approach

The proposed approach improves completeness of a BDAT test suite and enables coverage-based test sequence generation. With the assumption that Gherkin clauses can be represented by events, the proposed approach suggests the use of ESGs for modeling BDATs. To model a BDAT as an ESG, ESGs are extended with tags. This is explained first in this section. Then, how BDATs are combined using tagged ESGs is presented. After that, elimination of tags by combination process that is used to find missing BDATs is outlined. This section concludes with an example where all BDATs, i.e., original, missing, and additional BDATs, are composed into one ESG without any tags.

A. Representation of BDATs with tagged ESGs

Best practice for Gherkin scenarios is to describe behavior rather than functionality.

A behavior driven acceptance test is a specification of the behavior of the system, which verifies the interactions of the objects rather than their states [10]. A scenario that makes up a BDAT is composed of several steps. A step is an abstraction that represents one of the elements in a scenario which are: contexts, events, and actions [1]. So, a Gherkin scenario template is as follows:

Given context
When event
Then action

Contexts, events, and actions can be represented by events. A context is formed after a sequence of events. For instance, the line Given I am on the homepage in a scenario indicates that the context is being on the homepage and the user can reach the homepage by a sequence of events. So, we can say that a context is the result of a sequence of events. Sometimes, the sequence of events may be empty. An action is an event or results in an event depending on your standpoint. For instance, the line Then product list is displayed in a scenario is the action of the software, but for the user it is an event.

This paper proposes the use of event sequence graphs for modeling BDATs. To model a BDAT as an ESG, ESGs are extended with tags.

Definition 4. A tagged ESG is an ESG, where a node or vertex may contain a tag instead of an event.

A tagged ESG is useful in transforming Gherkin scenarios or BDATs to ESGs. Contexts and actions are represented by tags and this way, tags become connection or composition points for ESGs. For instance, in the following Scenario cart02, *Given* event is tagged with #productPage and *Then* event is tagged with #shoppingBasket. Its ESG representation is shown in Figure 2.

Scenario: cart02 - Adding a product to cart
Given I am on a product detail page #productPage
When I select the amount
And I click the add to cart button
Then the product is added to my shopping cart #shoppingCart

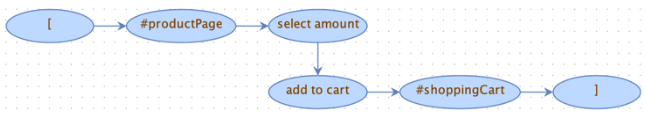


Figure 2. Tagged ESG for Scenario cart02.

Annotating Gherkin clauses with tags and representing BDATs with tagged ESGs enable us to combine BDATs.

B. *Combining two BDATs on tagged ESG*

To combine two BDATs, the following approach is proposed. Ending Gherkin clause can be combined with starting Gherkin clause if they have the same tag. This means two Gherkin scenarios can be run in a sequence. We can connect Scenario cart02 with Scenario check01 presented below, where *Given* event is tagged with #shoppingBasket and *Then* event is tagged with #orderConfirmed. ESG representation of Scenario check01 is shown in Figure 3.

Scenario: check01 - Successful checkout
Given I have added an item to my shopping bag #shoppingCart
When I proceed to the check out
And I enter valid delivery details
And I select a payment method
And I confirm the order
Then I am redirected to the thank you page #orderConfirmed

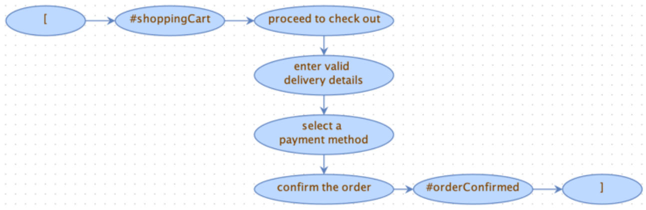


Figure 3. Tagged ESG for Scenario check01.

As seen, tags are used as connection points. Following the approach presented in Section III-A, we can combine these two BDATs on a tagged ESG, since both are represented as a tagged ESG. The resulting tagged ESG is shown in Figure 4.

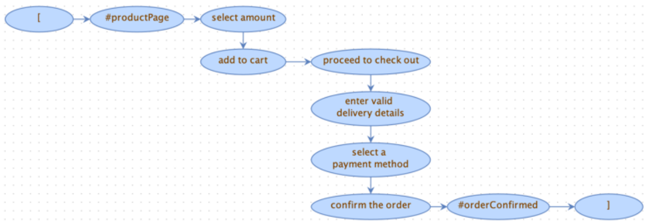


Figure 4. Tagged ESG for combined Scenarios cart02 and check01.

C. *Finding missing BDATs*

To find missing BDATs, elimination by combination is proposed. As seen in Section III-B, once two BDATs are combined using a tag, that tag is eliminated. Therefore, first all possible tagged scenarios or their graphical representations, i.e., tagged ESGs, are combined. It should be noted that a combined tagged ESG may be combined with another simple or combined tagged ESG. The goal is to reach an ESG without any tags, as shown in Figure 5. After all possible combinations are completed, if a tag remained on a tagged ESG indicates that there is a missing BDAT. If there are more than one tag, that may mean more missing BDATs.

For instance, in the following Scenario acc03, *Given* event is tagged with #atHome and *Then* event is tagged with #orderDetail.

Scenario: acc03 - Check orders
Given I am logged in on the site #atHome
When I navigate to my orders
Then I see a list of my orders
And I can open an order to see the order details #orderDetail

This BDAT is the only Gherkin scenario that has the tag #orderDetail. Since there is no match, it indicates that a BDAT that starts with #orderDetail tag is missing. We can complete this missing BDAT as follows:

Scenario: acc10 - Back to order list page
Given #orderDetail
When I press OK button
Then order list page is displayed #orderList

As seen in the running example, elimination by combination shows us clues about completeness of BDATs. The approach proposed here is to check whether all tags are combined. Any tag that is not combined suggests a missing BDAT.

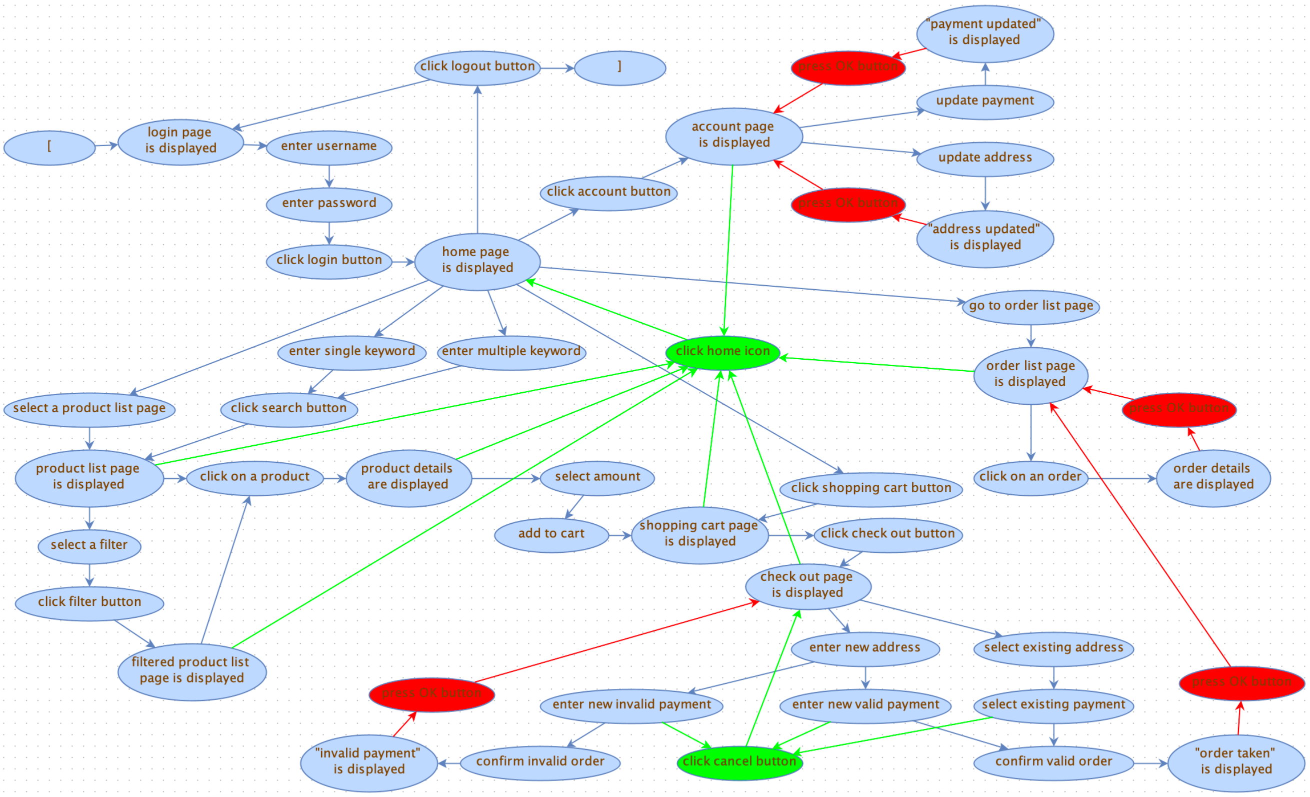


Figure 5. Composed ESG.

D. *Composition of BDATs on tagged ESG*

After completing the missing BDATs and improving existing BDATs, the BDATs are composed on an ESG. The resulting ESG is shown in Figure 5. There are no tags on the resulting ESG, which means that all tags can be eliminated by combination. Elimination by combination enables us to find five missing BDATs, which are drawn in red on the resulting ESG in Figure 5.

Once ESG is ready then CES for edge and for edge-pair coverage can be generated for BDATs. The details of CES generation can be found in [8]. We utilized the TSD tool [11] to generate CES for both coverage criteria. The results are given in the following section.

IV. Evaluation

For evaluation, the proposed approach is applied to an existing test suite for an e-commerce software [12], of which six features out of eight are taken for evaluation. The features locale and newsletter are left. The existing test suite has 15 scenarios, or BDATs, with 64 Gherkin clauses. Clause per scenario ratio is 4.26.

After applying the proposed approach, we end up with 24 BDATs and 85 Gherkin clauses. There are 9 new scenarios but only 5 of them are missing scenarios. The other 4 scenarios are introduced to simplify and standardize some original scenarios. So, clause per scenario ratio is decreased to 3.54 from 4.26. The comparison of before and after the proposed approach is given in Table I. The resulting test suite has the scenarios that are simplified, standardized, and tagged. Moreover, they become composable.

TABLE I. Comparison of Before and After Proposed Approach

Criteria	*Before*	*After*
Number of scenarios	15	24
Number of clauses	64	85
Clause per scenario ratio	4.26	3.54

A further analysis of the resulting ESG shows that event sequences are stuck in the child pages of home page. There is no return to home page from child pages, which means that features of the software cannot be tested in sequence. In addition, it is discovered that there is no scenario about cancellation of the check-out process. Those BDATs, 10 in total, are added in green to the resulting ESG in Figure 5. It should be noted that the graphical representation of BDATs enables us to perform such an analysis. Without tool support, it is very hard for test designers to conduct such analysis on text represented BDATs.

There is another advantage of the proposed approach. Since BDATs are transformed to ESGs and then combined, we have an ESG from which we can automatically generate test sequences, i.e., sequences of BDATs. CES for edge coverage computed by the TSD tool is shown below:

No. of Nodes: 50
No. of Edges: 70
CES with 111 events:

[, login page is displayed, enter username, enter password, click login button, home page is displayed, go to order list page, order list page is displayed, click on an order, order details are displayed, press OK button, order list page is displayed, click home icon, home page is displayed, click shopping cart button, shopping cart page is displayed, click check out button, check out page is displayed, enter new address, enter new invalid payment, confirm invalid order, "invalid payment" is displayed, press OK button, check out page is displayed, enter new address, enter new invalid payment, click cancel button, check out page is displayed, enter new address, enter new valid payment, click cancel button, check out page is displayed, select existing address, select existing payment, click cancel button, check out page is displayed, enter new address, enter new valid payment, confirm valid order, "order taken" is displayed, press OK button, order list page is displayed, click home icon, home page is displayed, enter multiple keyword, click search button, product list page is displayed, select a filter, click filter button, filtered product list page is displayed, click on a product, product details are displayed, select amount, add to cart, shopping cart page is displayed, click home icon, home page is displayed, enter single keyword, click search button, product list page is displayed, click on a product, product details are displayed, click home icon, home page is displayed, select a product list page, product list page is displayed, click home icon, home page is displayed, click account button, account page is displayed, update payment, "payment updated" is displayed, press OK button, account page is displayed, update address, "address updated" is displayed, press OK button, account page is displayed, click home icon, home page is displayed, click shopping cart button, shopping cart page is displayed, click check out button, check out page is displayed, select existing address, select existing payment, confirm valid order, "order taken" is displayed, press OK button, order list page is displayed, click home icon, home page is displayed, select a product list page, product list page is displayed, select a filter, click filter button, filtered product list page is displayed, click home icon, home page is displayed, click shopping cart button, shopping cart page is displayed, click check out button, check out page is displayed, click home icon, home page is displayed, click logout button, login page is displayed, enter username, enter password, click login button, home page is displayed, click logout button,],

CES for edge-pair coverage computed by the TSD tool has a complete event sequence of 224 events. The CES is not given here because of space limitations.

V. Discussion

The proposed approach assumes that Gherkin clauses can be represented by events. This assumption holds for the selected test suite used in the evaluation. Although Gherkin is developed for behavioral description scenarios, it must be shown that all possible Gherkin clauses and scenarios can be represented by events. It may require some transformation. This is left as future work.

The proposed approach shows that through modeling BDATs, it is possible to automatically generate test sequences. UML use case diagrams and activity diagrams can also be used for modeling BDATs and then automatically generate tests. The research in this area is explained in the related work section.

Scalability of the models is an important concern. ESGs allow us to work on some small and modular models through sub-ESGs [6]-[9] like subroutines. The TSD tool is also designed to support sub-ESGs. This way, it is possible to generate manageable large models. Moreover, these sub-ESGs can be flattened into one large ESG if necessary.

One threat to validity is internal validity, which deals with the effects on the evaluation. The selection of BDAT test suite used in evaluation is obtained by searching GitHub repositories. This cannot be considered as random selection. Moreover, the proposed approach is applied to the selected BDAT test suite by the author.

Another threat to validity is external validity, which deals with the generalizability of the results. The evaluation in this study is based on a single BDAT test suite. Although this test suite is developed for e-commerce software, which may represent business software generally, evaluation of other BDAT test suites from different domains with the proposed approach will help generalize the results.

VI. Related Work

Tuglular [13] proposed a model-based approach for feature-oriented testing using Event Sequence Graphs (ESGs). In this approach, ESGs are extended to save state and pass it to the following ESG. This way, tests written for features can be combined on state information. However, capturing state is not always possible for acceptance tests.

UML use case diagrams can also be used for modeling BDATs and then automatically generate tests. Gutierrez et al. [14] proposed an approach for working with Gherkin scenarios using UML use case models. They transform from the UML use case diagrams to the Gherkin plain text syntax. They also developed a tool for running Gherkin scenarios in UML as test cases.

Alferez et al. [15] proposed an approach, named AGAC (Automated Generation of Acceptance Criteria), which supports the automated generation of AC specifications in Gherkin. They used UML use case diagrams and activity diagrams to create specifications, derive acceptance criteria from them, and then generate test cases from derived acceptance criteria.

Kudo et al. [16] proposed the software pattern meta model that bridges requirement patterns to groups of scenarios with similar behaviors in the form of test patterns. This meta model is used to describe the behavior of a requirement pattern through a time executable and easy-to-use language aiming at the automatic generation of test patterns.

Wanderley and da Silveria [17] proposed using a mind model specification, which serves as a basis for transforming the definitions of the scenario and generating a conceptual model represented by a UML class diagram. The mind model functions as a bond that represents the business entities, and enables simple association, aggregation and composition relationships between the entities.

An adjacent area is process discovery in business process management literature. Rozinat and van der Aalst [18] worked on whether event logs conform to the process model and vice versa. They proposed two dimensions of conformance, namely fitness and appropriateness, to be checked along with corresponding metrics. They developed a Conformance Checker within the ProM Framework.

Beschastnikh et al. [19] proposed algorithms for inferring communicating finite state machine models from traces of concurrent systems, and for proving them correct. They also provided an implementation called CSight, which helps developers find bugs.

Pecchia et al. [20] proposed an approach that employs process mining for detecting failures from application logs. Their approach discovers process models from logs; then it uses conformance checking to detect deviations from the discovered models. They were able to quantify the failure detection capability of conformance checking in spite of missing events, and its accuracy with respect to process models obtained from noisy logs [20].

VII. Conclusion

This paper proposes an approach to represent BDATs using ESGs. With the proposed approach, the test designer not only finds and completes missing BDATs, but also combines them to know which BDAT can be executed after which BDAT. When the final composition is supplied to the TSD tool, it automatically generates a test sequence that covers all BDATs. So, the proposed approach improves testability of BDATs.

As future work, we plan to automate the processes explained here and develop a tool. Also as future work, our goal is to enhance the tool with ontologies so semantically related scenarios are easily decoded.

References

[1] M. G. Cavalcante and J. I. Sales, "The Behavior Driven Development Applied to the Software Quality Test," Proc. 14th Iberian Conference on Information Systems and Technologies (CISTI), IEEE, 2019, pp.1–4.

[2] Cucumber Gherkin. https://cucumber.io/docs/gherkin/reference/. [retrieved: March, 2021].

[3] Gauge. https://gauge.org. [retrieved: March, 2021].

[4] T. Tuglular and S. Şensülün. "SPL-AT Gherkin: A Gherkin Extension for Feature Oriented Testing of Software Product Lines." 2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC). Vol. 2. IEEE, 2019, pp.344–349.

[5] T. Tuglular, F. Belli, and M. Linschulte, "Input contract testing of graphical user interfaces," International Journal of Software Engineering and Knowledge Engineering, 26(02), 2016, pp.183–215.

[6] F. Belli and C. J. Budnik, "Test minimization for human-computer interaction," Applied Intelligence, 26(2), 2007, pp.161–174.

[7] F. Belli, C. J. Budnik, and L. White, "Event based modelling, analysis and testing of user interactions: approach and case study," Software Testing, Verification and Reliability, 16(1), 2006, pp.3–32.

[8] F. Belli and C. J. Budnik, "Minimal spanning set for coverage testing of interactive systems," International Colloquium on Theoretical Aspects of Computing. Springer, Berlin, Heidelberg, 2004, pp.220–234.

[9] T. Tuglular, C. A. Muftuoglu, F. Belli, and M. Linschulte, "Event-based input validation using design-by-contract patterns," Proc. 20th International Symposium on Software Reliability Engineering, ISSRE'09, IEEE Press, 2009, pp. 195–204.

[10] E. Evans, "Domain-Driven Design: Tackling Complexity in the Heart of Software," Addison-Wesley Professional, 2003.

[11] TestSuiteDesigner. http://download.ivknet.de/. [retrieved: March, 2021].

[12] Barzilay, "Example of an ECommerce cucumber web test automation suite". https://github.com/spriteCloud/ecommerce-cucumber-web-test-automation-suite. [retrieved: March, 2021].

[13] T. Tuglular, "Event sequence graph-based feature-oriented testing: A preliminary study," 2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C). IEEE, 2018, pp. 580–584.

[14] J. J. Gutiérrez, I. Ramos, M. Mejías, C. Arévalo, J. M. Sánchez-Begines, and D. Lizcano, "Modelling Gherkin Scenarios Using UML," Proc. 26th International Conference on Information Systems Development (ISD), 2017, http://aisel.aisnet.org/isd2014/proceedings2017/ISDMethodologies/7.

[15] M. Alferez, F. Pastore, M. Sabetzadeh, L. Briand, and J. R. Riccardi, "Bridging the gap between requirements modeling and behavior-driven development," 2019 ACM/IEEE 22nd International Conference on Model Driven Engineering Languages and Systems (MODELS), IEEE, 2019, pp. 239–249.

[16] T. N. Kudo, R. F. Bulcão-Neto, and A. M. Vincenzi, "A conceptual metamodel to bridging requirement patterns to test patterns," Proc. of the XXXIII Brazilian Symposium on Software Engineering. 2019, pp.155–160.

[17] F. Wanderley and D. S. da Silveria, "A framework to diminish the gap between the business specialist and the software designer," 2012 Eighth International Conference on the Quality of Information and Communications Technology. IEEE, 2012, pp. 199–204.

[18] A. Rozinat and W.M.P. van der Aalst, "Conformance testing: Measuring the fit and appropriateness of event logs and process models," Proc. 4th Business Process Management Workshops, Springer, 2006, pp. 163–176.

[19] I. Beschastnikh, Y. Brun, M.D. Ernst, and A. Krishnamurthy, "Inferring models of concurrent systems from logs of their behavior with CSight," Proc. 36th International Conference on Software Engineering, ACM, 2014, pp. 468–479.

[20] A. Pecchia, I. Weber, M. Cinque, and Y. Ma., "Discovering process models for the analysis of application failures under uncertainty of event logs," Knowledge-Based Systems, 2020, 189: 105054, https://doi.org/10.1016/j.knosys.2019.105054.