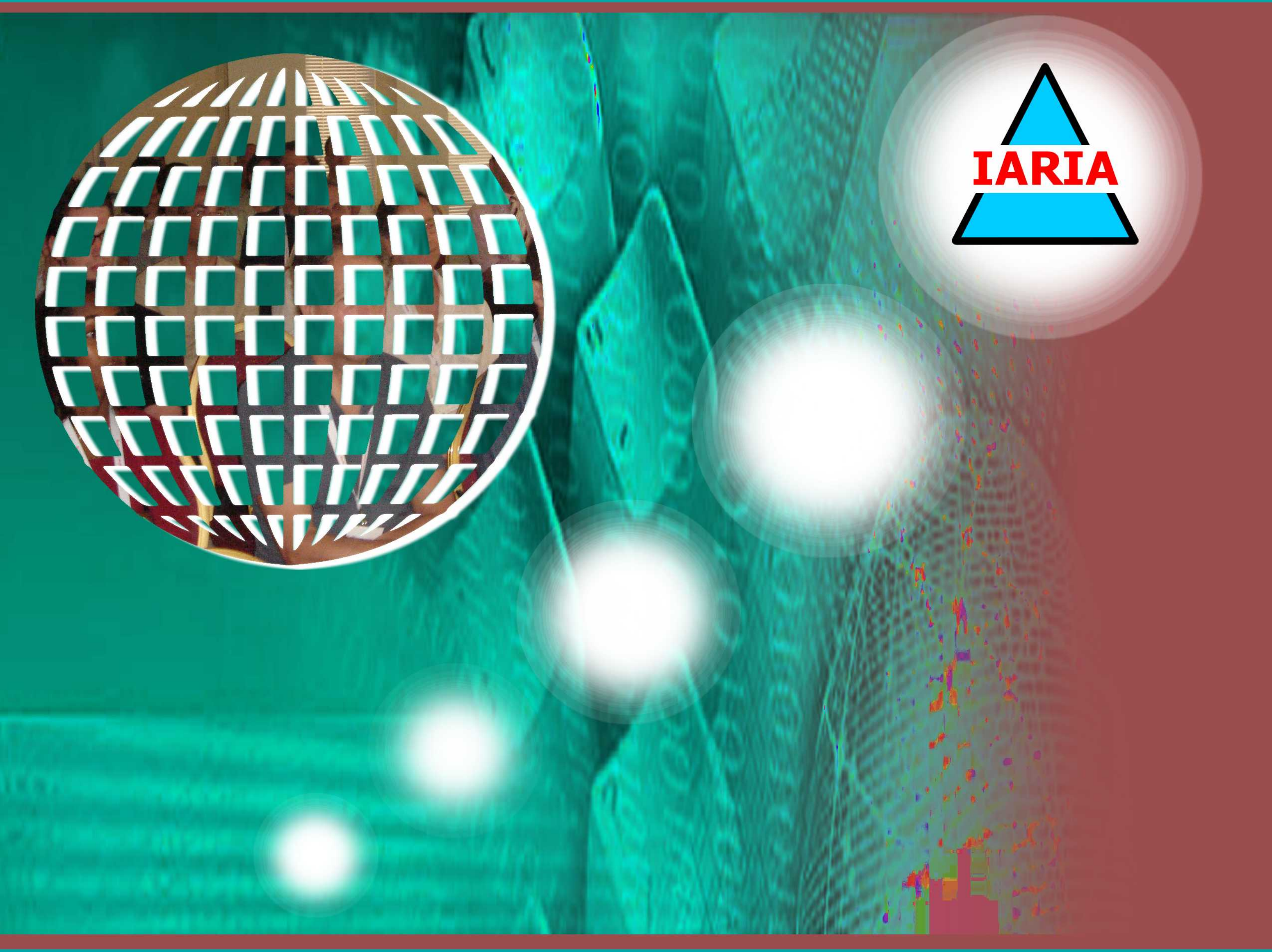
International Journal on Advances in Software



2024 vol. 17 nr. 1&2

The *International Journal on Advances in Software* is published by IARIA.
ISSN: 1942-2628
journals site: http://www.iariajournals.org
contact: petre@iaria.org

Responsibility for the contents rests upon the authors and not upon IARIA, nor on IARIA volunteers, staff, or contractors.

IARIA is the owner of the publication and of editorial aspects. IARIA reserves the right to update the content for quality improvements.

Abstracting is permitted with credit to the source. Libraries are permitted to photocopy or print, providing the reference is mentioned and that the resulting material is made available at no cost.

Reference should mention:

International Journal on Advances in Software, issn 1942-2628
vol. 17, no. 1 & 2, year 2024, http://www.iariajournals.org/software/

The copyright for each included paper belongs to the authors. Republishing of same material, by authors or persons or organizations, is not allowed. Reprint rights can be granted by IARIA or by the authors, and must include proper reference.

Reference to an article in the journal is as follows:

<Author list>, "<Article title>"
International Journal on Advances in Software, issn 1942-2628
vol. 17, no. 1 & 2, year 2024,<start page>:<end page> , http://www.iariajournals.org/software/

IARIA journals are made available for free, proving the appropriate references are made when their content is used.

Sponsored by IARIA
www.iaria.org

Copyright © 2024 IARIA

International Journal on Advances in Software
Volume 17, Number 1 & 2, 2024

Editor-in-Chief

Petre Dini, IARIA, USA

Editorial Board

Chrissanthi Angeli, University of West Attica, Greece
Vincenzo Arceri, University of Parma, Italy
Thierry Badard, Université Laval, Canada
Fabian Barbato, UDELAR, Uruguay
Fernando Boronat Segui, Universitat Politecnica de Valencia, Spain
Mina Boström Nakićenović, Flightradar24, Stockholm, Sweden
Antonio Brogi, University of Pisa, Italy
Carlos Casanova, UTN-FRCU, Argentina
Yoonsik Cheon, The University of Texas at El Paso, USA
Stefano Cirillo, University of Salerno, Italy
Emanuele Covino, Università di Bari, Italy
Mirela Danubianu, "Stefan cel Mare " University of Suceava, Romania
Cláudio de Souza Baptista, Federal University of Campina Grande, Brazil
Maria del Pilar Angeles, Universidad Nacional Autonónoma de México, México
Rafael del Vado Vírseda, Universidad Complutense de Madrid, Spain
Leandro Dias da Silva, Universidade Federal de Alagoas, Brazil
Roland Dodd, CQUniversity, Australia
Ann Dunkin, US Department of Energy, USA
Gledson Elias, Federal University of Paraíba (UFPB), Brazil
Thomas Fehlmann, Euro Project Office, Switzerland
Stoyan Garbatov, OutSystems, Portugal
Apostolos Gkamas, University of Ioannina, Greece
Gregor Grambow, Aalen University, Germany
Christoph Grimm, Kaiserslautern University of Technology, Germany
Andreas Hausotter, University of Applied Sciences and Arts, Hanover, Germany
Tzung-Pei Hong, National University of Kaohsiung, Taiwan
Chih-Cheng Hung, Kennesaw State University, USA
Dayang Norhayati Abang Jawawi, Universiti Teknologi Malaysia (UTM), Malaysia
Ahmed Kamel, Concordia College, Moorhead, USA
Ayad Ali Keshlaf, Sabratha University, Libya
Maqbool Khan, Pak-Austria Fachhochschule - Institute of Applied Sciences and Technology, Pakistan
Radek Koci, Brno University of Technology, Czech Republic
Dmitry Korzun, Petrozavodsk State University, Russia
Jana Kostičová, Comenius University Bratislava, Slovakia
Luigi Lavazza, Università dell'Insubria, Italy
Zheng Li, Queen's University Belfast, UK

Panos Linos, Butler University, USA
Qifeng Lu, Pivotal Commware, USA
Herwig Mannaert, University of Antwerp, Belgium
Adriana Martin, Universidad Nacional de la Patagonia Austral, Argentina
Goran Martinovic, J. J. Strossmayer University of Osijek, Croatia
Paulo Martins, University of Trás-os-Montes e Alto Douro (UTAD), Portugal
José Carlos M. M. Metrolho, Polytechnic Institute of Castelo Branco, Portugal
Jose Manuel Molina Lopez, Universidad Carlos III de Madrid, Spain
Fernando Moreira, REMIT, Universidade Portucalense, Portugal
Roy Oberhauser, Aalen University, Germany
Constantin Paleologu, National University of Science and Technology Politehnica Bucharest, Romania
Elzbieta Pustulka, University of Applied Sciences and Arts Northwestern Switzerland (FHNW), Switzerland
Kornelije Rabuzin, University of Zagreb, Croatia
Piotr Ratuszniak, Koszalin University of Technology, Poland
Hajarisena Razafimahatratra, Ecole Nationale d'Informatique - Université de Fianarantsoa, Madagascar
José Rouillard, University of Lille, France
Claus-Peter Rückemann, Universität Münster / DIMF / Leibniz Universität Hannover, Germany
Sébastien Salva, IUT Clermont Auvergne | University of Clermont Ferrand, France
Patrizia Scandurra, Università degli Studi di Bergamo, Italy
Mu-Chun Su, National Central University, Taiwan
Maryam Tayefeh Mahmoudi, ICT Research Institute, Iran
Mónica Isabel Teixeira da Costa, Technology School | Polytechnic Institute of Castelo Branco, Portugal
Pierre Tiako, Langston University, USA
Božo Tomas, University of Mostar, Bosnia and Herzegovina
Mariusz Trzaska, Polish-Japanese Academy of Information Technology, Poland
Chrisa Tsinaraki, Technical University of Crete, Greece
Miroslav Velev, Aries Design Automation, USA
Mudasser F. Wyne, National University, USA
Martin Zinner, Technische Universität Dresden, Germany

International Journal on Advances in Software
Volume 17, Numbers 1 & 2, 2024

CONTENTS

pages: 1 - 12
Insights into the Grammar and Real User Connectivity for Choreography Patterns of Microservices-based Insurance Processes
Christin Schulze, Hochschule Hannover, Germany
Alexander Link, Hochschule Hannover, Germany
Andreas Hausotter, Hochschule Hannover, Germany
Arne Koschel, Hochschule Hannover, Germany

pages: 13 - 22
Digital Wallets and Identity Management: Pioneering Advances for Cloud Service Evolution
Fatemeh Stodt, IDACUS, Furtwangen University, Germany
Christoph Reich, IDACUS, Furtwangen University, Germany

pages: 23 - 32
External Representations of Semantically Rich Content in Complex Systems Using Loose Coupling
Hans-Werner Sehring, Nordakademie, Germany

pages: 33 - 43
Reliable Color Recognition in Images by Using a Modified HSV Algorithm
Samuel Kosolapov, Braude College of Engineering, Israel

pages: 44 - 58
The Project Management Assistant that Never Sleeps - Using Generative Artificial Intelligence to Support Project Management Activities
Claudia Hess, IU International University of Applied Sciences, Germany
Sibylle Kunz, IU International University of Applied Sciences, Germany

pages: 59 - 67
Verifiable Labels for Digital Services: A New Approach to Phishing Detection
Maël Gassmann, Bern University of Applied Sciences, Switzerland
Annett Laube, Bern University of Applied Sciences, Switzerland

pages: 68 - 79
A Multivocal Review on Derivation Games: A Software Reuse Study
Diego Castro, Universidade do Estado do Rio de Janeiro, Brazil
Cláudia Werner, Universidade do Estado do Rio de Janeiro, Brazi

pages: 80 - 89
Model-Supported Software Creation: Extending Model-Driven Software Engineering with Non-Formal Artifacts and Transformations
Hans-Werner Sehring, Nordakademie, Germany

pages: 90 - 99
An Integrated Software Manufacturing Control System for a Software Factory with Built-In Rejuvenation
Herwig Mannaert, University of Antwerp, Belgium

Jan Verelst, University of Antwerp, Belgium
Koen De Cock, NSX bv, Belgium
Jeroen Faes, NSX bv, Belgium

pages: 100 - 115
Regional Feature Importance for Error Analysis in Manufacturing
Valentin Göttisheim, Furtwangen University, Germany
Holger Ziekow, Furtwangen University, Germany
Peter Schanbacher, Furtwangen University, Germany

pages: 116 - 128
Transformation of Entity Relationship Models to Relational Models - A Practical Approach
Gregor Grambow, Aalen university, Germany
Simon Ruttmann, eXXcellent solutions GmbH, Germany

pages: 129 - 142
Attempt to Design Electronic Bulletin Board Based on Information-Sharing Behavior Observations with Analog Bulletin Boards
Kanayo Ogura, Iwate Prefectural University, Japan
Ryotaro Hoshi, QUICK CO., LTD., Japan

Insights into the Grammar and Real User Connectivity for Choreography Patterns of Microservices-based Insurance Processes

Arne Koschel
Andreas Hausotter
Hochschule Hannover
University of Applied Sciences & Arts Hannover
Faculty IV, Department of Computer Science
Hanover, Germany
Email: {arne.koschel, andreas.hausotter}
@hs-hannover.de

Christin Schulze
Alexander Link
Hochschule Hannover
University of Applied Sciences & Arts Hannover
Faculty IV, Department of Computer Science
Hanover, Germany
Email: {christin.schulze, alexander.link}
@stud.hs-hannover.de

Abstract—**To avoid the shortcomings of traditional monolithic applications, the Microservices Architecture (MSA) style plays an increasingly important role in providing business services. This is especially true for the insurance industry with its sophisticated cross-domain business processes. Here, the question arises of how workflows can be implemented to grant the required flexibility and agility and, on the other hand, exploit the MSA style's potential. There are two competing approaches to workflow realization, orchestration, and choreography, each with pros and cons. Though choreography seems to be the method of choice in MSA, it comes with some challenges. As the workflow is implicit – it evolves as a sequence of events being sent around – it becomes difficult to understand, change, or operate the workflow. To manage the challenges of the choreography approach, we use BPMN 2.0 choreography diagrams to model the exchange of domain events between microservices, which represent 'participants' in terms of BPMN. We aim to execute choreography diagrams automatically. For this, we developed a set of choreography patterns that represent frequently occurring sequences. We present the pattern language and discuss five patterns, a One-Way Task pattern, a Two-Way Task pattern, an Event-based Gateway – Deadline pattern, and an Open Parallel Gateway – different Senders pattern. Additionally, we present how a real user may get integrated into these interactions and provide insights into a grammar, which can validate the order of use for patterns in a given choreography. This paper is part of our ongoing research to design a microservices reference architecture for insurance companies.**

Keywords—Workflow; Choreography; BPMN; Patterns; Grammar; Business Processes; Microservice.

I. INTRODUCTION

In this article, which is an extended version of our earlier paper [1], we look at the realization of interactions within a microservices-based reference architecture for German insurance companies. Business workflows and multistep business processes are typical for insurance companies; see, for example, the reference architecture for German insurance companies (VAA) [2]. They are complemented by general regulations, such as the European GDPR [3], as well as insurance-specific laws and rules regarding, for example, financial regulations, data protection, and security [4].

Recently, the Microservices Architecture (MSA) style [5] [6] and cloud computing [7] became more and more interesting for insurance companies. Traditionally, several technologies from monolithic mainframe applications, functional decomposition-based software, traditional Service-Oriented Architectures (SOAs), which often utilize some kind of Enterprise Service Bus (ESB), Business Process and Workflow Management Systems (BPMS, WfMS) for orchestration, and 3rd party software, such as SAP software, were and are used together in insurance business applications, which implement their business processes.

Taking all those typical cornerstones from (over time grown) insurances into account, the goal of our currently ongoing research [8] is to develop a 'Microservice Reference Architecture for Insurance Companies (RaMicsV)' jointly with partner companies from the insurance domain. Within our work, we also look at the question: 'how to implement (insurance) business workflows with microservices, which potentially utilize several logical parts from RaMicsV'?

Within the MSA style, the more decoupled choreography is favored for this purpose [5] [6]. This is in some contrast, however, for example, to SOAs, where such workflows are mainly implemented using orchestration [9]. For example, one of our partner companies utilizes Camunda [10], another one a Java/Jakarta EE-based workflow tool.

However, since co-existence of all approaches is a 'must have' for our insurance partner companies, RaMicsV aims to address the *combined usage* of more traditional approaches and the MSA style, the combination of choreography and orchestration naturally comes to mind. As evolution is a key demand for our business partners – they can and will not just 'throw away' their existing application landscape – concepts such as orchestration and tools such as an ESB, whose use within MSA style architectures are both clearly disputable, have to be integrated reasonably well into our approach.

We thus started to look at the *combination* of choreography and orchestration, including a look at insurance domain specifics, in our work from [11]. In the present article, we will now have a focus on choreography-based approaches for (insurance) business processes. Particularly, we will examine an initial set of emerged *choreography patterns* for this pur-

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

pose, which we will model using choreography diagrams from the OMG BPMN 2.0 standard [12]. It should be noted that our goal is not a general implementation of choreographies, rather an implementation that orients itself toward real-world scenarios. Thus, we inspected multiple use cases from the insurance industry, one of which we will introduce later on.

In particular, we contribute in the present article our ongoing work and intermediate results about:

- The integration of the choreography within our RaMicsV;
- BPMN 2.0 choreography diagrams and the utilization of patterns;
- our pattern language for choreography patterns;
- four particular choreography patterns in depth, namely the One-Way Pattern, Two-Way pattern Open-Parallel Gateway – Different Senders pattern and the Event-based Gateway – Deadline pattern;
- User Connectivity within these interactions;
- insights into a grammar, to validate those patterns;
- and finally, insurance business use cases for those patterns.

The remainder of this article is structured as follows: After discussing related work in Section II, we briefly look at our current work within the RaMicsV context in Section III. Next, Section IV looks at BPMN 2.0 choreography diagrams with patterns. Section V then contributes our patterns usage and a pattern language for them, as well as four identified patterns, two of which are new to this version. Afterward, Section VI shows a possible way of integrating real users into these choreographies. Then, the current state of the choreography grammar will be presented in Section VII.

Moreover, Section VIII looks at a usage of those patterns within an insurance business use case. Finally, Section IX summarizes our results and concludes with some outlook to future work, with more patterns to follow.

II. RELATED WORK

The basis of our research builds on authors in the scope of microservices, such as the work from Newman [6], as well as Fowler and Lewis [13]. Within the design of our reference architecture, we profit from different microservices patterns, as they are discussed by Krause [14] and Richardson [5].

To model our business processes, we use OMG's BPMN 2.0 specification. Also, we use as groundwork about business processes and its development with BPMN the works from Allweyer [15] [16], Rücker and Freund [17].

For the basics of service composition types, orchestration and choreography, we chose to rely on Decker's approach [18]. It is important that we define the choreography in terms of workflows within a microservices architecture. Quite many publications discuss the benefits of the choreography as a composition between (micro-)services. In particular, in several cases the theoretical benefit is presented or the combination of different approaches with the choreography is shown, as discussed by Rücker in his blog [19].

This paper ties in with our previous work on realizing a choreography [11]. In our last paper, we experimented with the implementation of a choreography using BPMN. The first pattern 'Any Problem becomes a Service' appeared to be difficult, since the monolithic BPMN does not support the message exchange between different microservices.

In Mikalkinas' [20] approach, a BPMN choreography diagram is transformed into a BPMN collaboration diagram and then executed. After this transformation, the BPMN collaboration diagram is executed by an engine, in this case Camunda [10]. We intend to bypass this conversion and provide direct execution of the choreography diagram. Thereby, our goal is to explore an implementation without an engine, since this corresponds to an orchestration in the case of Camunda.

Milanović and Gasević also try to implement choreography via BPMN and REWERSE II Rule Markup Language in their work [21]. They developed a rule-based extension for BPMN to realize choreography, called rBPMN. Ortiz et al. describe a similar approach [22]: In their work, rules are also defined on how to react based on which events in a choreography. This work uses fragments of BPMN. In both approaches (only) parts of the BPMN are considered, and in each case, only collaboration diagrams.

Another related approach is Richardson's SAGA pattern and the Eventuate Framework [5] [23]. The pattern describes the splitting of a transaction into several small local transactions. The local transactions trigger each other by messages/events. The error handling could become interesting for our further work. The framework includes two manifestations: Tram and Local. Eventuate Tram [24] so far only implements an orchestrated SAGA, so it does not yet include a choreography. Eventuate Local [25] provides event sourcing to store events. It also offers functions to perform transactions, through a publish/subscribe realization. It maps the technical implementation of a transaction rather than the communication and composition between services.

We try to implement a choreography in a more straight way as a compositional approach between microservices. Our vision is to use the choreography for the complete communication and workflow. We define the choreography as a global approach to processing a workflow without the intervention of a controlling part. This approach was described by us in our previous paper [11] and is also defined by Decker [18].

In order to evaluate interactions created from the choreography patterns using BPMN 2.0 choreography, we have created a grammar. For the creation of the grammar, we have used works by Vossen and Witt [26] and Priese and Erk [27] for advice.

To achieve this goal, we define patterns for BPMN choreography diagrams, which are supposed to be implemented automatically. To model our BPMN choreography diagrams, we used the framework chor-js developed by Ladleif et al. [28].

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

We do not focus on the processes within the (micro-) services themselves, rather only on the communication between them and the infrastructure. The use of patterns should also mitigate to some degree the complexity that can arise in (extensive) choreography-based workflows. The developed patterns borrow in structure and approach from Barros et al. [29].

III. Service-based Reference Architecture for Insurance Companies

This section presents our logical reference architecture for microservices in the insurance industry (RaMicsV) as initially started in [8].

RaMicsV defines the setting for the architecture and the design of a microservices-based application for our industry partners. The application's architecture will only be shown briefly, as it heavily depends on the specific functional requirements.

When designing RaMicsV, a wide range of restrictions and requirements given by the insurance company's IT management have to be considered. Regarding this contribution, the most relevant are:

- Enterprise Service Bus (ESB): The ESB as part of the SOA must not be questioned. It is part of a successfully operated SOA landscape, which seems suitable for our industry partners for several years to come. Thus, from their perspective, the Microservices Architecture (MSA) style is only suitable as an additional enhancement and only a partial replacement of parts from their SOA or other self-developed applications.
- Coexistence: Legacy applications, SOA, and microservices-based applications will be operated in parallel for an extended transition period. This means that RaMicsV must provide approaches for integrating applications from different architecture paradigms – looking at it from a high-level perspective, allowing an 'MSA style best-of-breed' approach at the enterprise architectural level as well.
- Business processes are critical elements in an insurance company's application landscape. To keep their competitive edge, the enterprise must change their processes in a flexible and agile manner. RaMicsV must therefore provide suitable solutions to implement workflows while ensuring the required flexibility and agility.

Figure 1 depicts the building blocks of RaMicsV which comprises layers, components, interfaces, and communication relationships. Components of the reference architecture are colored yellow; those out of scope are greyed out.

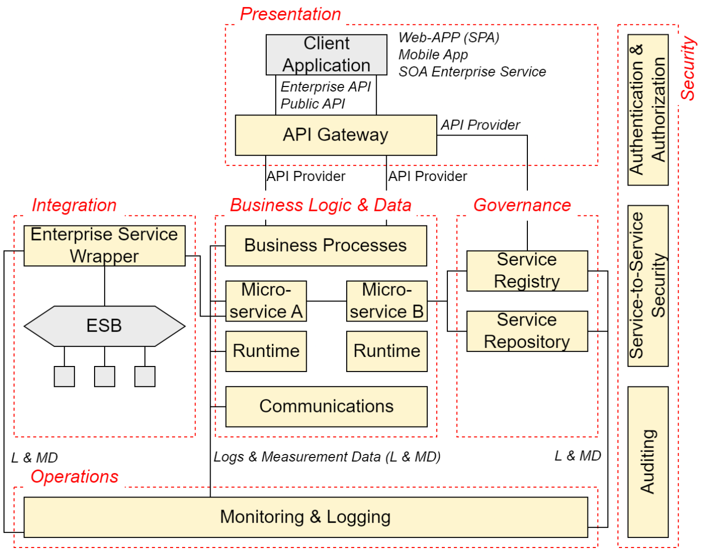


Figure 1. Building Blocks of the Logical Reference Architecture RaMicsV.

A component may be assigned to one of the following *responsibility areas*:

- **Presentation** includes components for connecting clients and external applications, such as SOA services.
- **Business Logic & Data** deals with the implementation of an insurance company's processes and their mapping to microservices, using various workflow approaches to achieve desired application-specific behavior.
- **Governance** consists of components that contribute to meeting the IT governance requirements of our industrial partners.
- **Integration** contains system components to integrate microservices-based applications into the industrial partner's application landscape.
- **Operations** consist of system components to realize unified monitoring and logging, which encloses all systems of the application landscape.
- **Security** consists of components to provide the goals of information security, i.e., confidentiality, integrity, availability, privacy, authenticity & trustworthiness, nonrepudiation, accountability, and auditability.

Components communicate either via HTTP(S) – using a RESTful API, or message-based – using a Message-Oriented Middleware (MOM) or the ESB. The ESB is part of the integration responsibility area, which itself contains a message broker (see Figure 1).

In the next section, we will take a look at the choreography in general and BPMN 2.0 choreography in particular as a lead-in to this paper's contribution, located in the responsibility area *Business Logic & Data*.

IV. Choreography

This section will present the core definition of choreography, as described in [11]. We briefly outline the use of BPMN, specifically the choice of BPMN 2.0 choreography diagrams.

A. *Choreography*

In a choreographed system, there exists no central coordinator, unlike in orchestration [30]. Decker [18] describes the definition of a choreography as a global view of how services

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

cooperate and the interaction between participants. This proves to be a challenge when modeling and monitoring a workflow, as the workflow is mapped by the interaction between the participants. It follows that the responsibility of executing and processing the workflow is transferred to each participant [31].

While choreography may be combined with other patterns, like the event-driven architecture [32], we decided not to focus on technical implementations yet, but will eventually.

B. BPMN 2.0 choreography

BPMN 2.0 choreography is chosen as the modeling language, since BPMN is also used by our partners. In the BPMN specification exist at least three significantly different diagram types to describe processes:

- **Process** known as classic BPMN. It visualizes the entire process.
- **Collaboration** splits a classic process into multiple participants (or microservices). Each sub-process in a participant can be recognized, but also the message exchange between the participants.
- **Choreography** which visualizes only the exchange of messages between participants.

In contrast to our previous work [11], we now focus only on the implementation of BPMN 2.0 choreography diagrams [12], as they visualize the interaction between microservices. In these diagrams, a participant represents a microservice. We aim to execute business processes using a choreographed MSA. Choreography serves as a global composition pattern [18]. We start with a collaboration diagram to map the whole process, which we then transform into a choreography diagram to focus on the communication. The processes within the participants are out of scope as we focus on the means of communication.

To automatically implement the choreography with BPMN 2.0 choreography, we develop patterns that map frequently occurring sequences. It should be a wide selection of things that must, should or can occur. The pattern language and the yet-to-be-developed grammar will be used to create a tool that automatically accepts modeled choreography diagrams and generates the necessary infrastructure and message exchange.

V. CHOREOGRAPHY PATTERNS

In this section, we will present a pattern language, as well as two patterns from our list. The language intends patterns to be assembled to produce more extensive use cases. The patterns originate from real-world use cases.

A. Pattern Language

A pattern language is utilized to describe the patterns uniformly. It consists of the following elements (cf. [7]):

- **Identification number (ID)** of the pattern.
- **Name** of the pattern.
- **Figures** that visualize the pattern. Consisting of BPMN 2.0 choreography diagrams, BPMN collaboration diagrams, and UML Sequence diagrams.
- A **Description** which describes the use, content, and flow of the pattern.
- **Rules** and conditions under which the pattern may be used.
- A list of **used BPMN elements** from the choreography- and collaboration diagrams, as named in [12].
- **Used Patterns**, which this pattern builds upon.
- **Synonyms** and similar patterns from literature and industry.
- **Variations** where the core concept of the pattern stays the same.
- **Typical combinations** and patterns with high compatibility.
- Example **Use-Cases** from the industry.

B. One-Way Task

Now that the pattern language has been introduced, we start with the most atomic pattern, the *One-Way Task*.

- **ID:** BPMNChor01
- **Name:** One-Way Task
- **Figures:** See Figure 2, Figure 3, and Figure 4.
- **Description:** Participant A wants to deliver a message to Participant B. The initiator (A) sends the message to the receiver (B).
- **Rules:** None.
- **Used BPMN Elements:** startEvent (none), messageStartEvent, participant (pool), Message originating from the initiator, endEvent (none).
- **Used Patterns:** None, this pattern is atomic and depicts the minimum amount of interaction.
- **Synonyms:** Fire-and-Forget, One-Way Notification
- **Variations:** None.
- **Typical combinations:** Due to the atomic properties of this pattern, it may be combined with every other pattern.
- **Use-Case:** Sending an E-Mail or push-notification. For a longer scenario, see Section VIII.

C. Two-Way Task

The *Two-Way Task* describes a bidirectional message exchange between two participants within a task.

- **ID:** BPMNChor02
- **Name:** Two-Way Task
- **Figures:** See Figures 5, 6, and 7.
- **Description:** Participant A initiates a message exchange with participant B. Participant A waits for participant B's response.
- **Rules:** None.
- **Used BPMN Elements:** startEvent (none), messageStartEvent, participant (pool), Message originating from the initiator, endEvent (none).

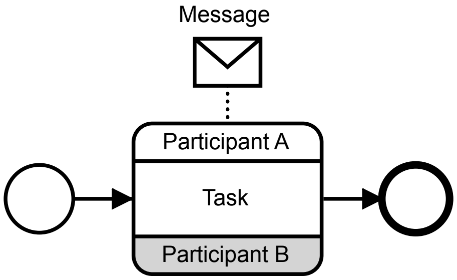


Figure 2. One-Way Task Choreography.

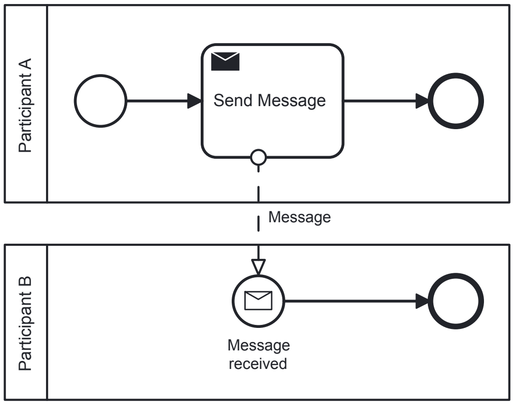


Figure 3. One-Way Task Collaboration.

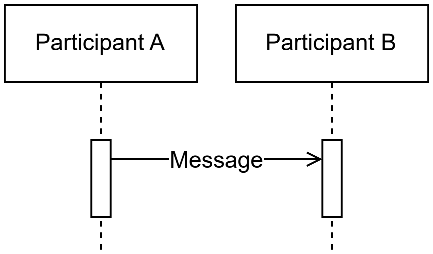


Figure 4. One-Way Task UML Sequence.

- **Used Patterns:** Two-Way Task is an extension of the One-Way Task. The extension adds the response from participant B.
- **Synonyms:** synchronous request-response.
- **Variations:** None.
- **Typical combinations:** Due to the atomic properties of this pattern, it may be combined with every other pattern.
- **Use-Case:** Message exchange via SMS. For a longer scenario, see Section VIII

This concludes the One-Way Task and the Two-Way Task as the minimal way of communication, next we will introduce Event-based Gateway – Deadline pattern.

D. Event-based Gateway – Deadline

The *Event-based Gateway – Deadline* pattern describes a more complex, yet often occurring, scenario where the flow of a process is determined by a temporal aspect.

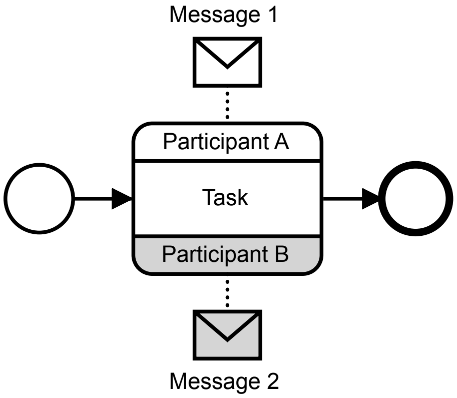


Figure 5. Two-Way Task Choreography.

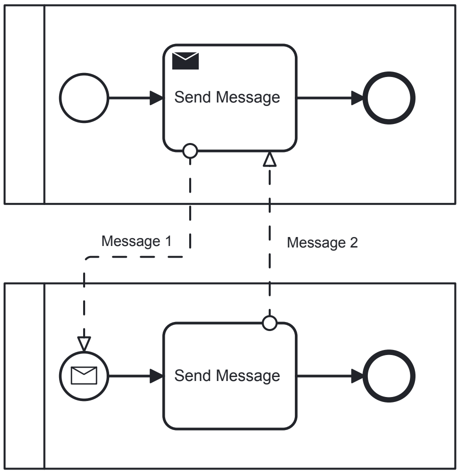


Figure 6. Two-Way Task Collaboration.

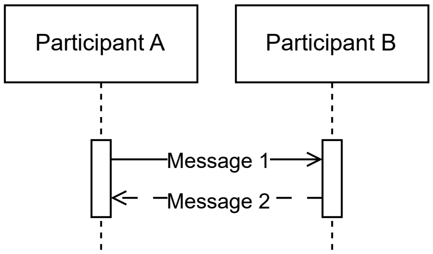


Figure 7. Two-Way Task UML Sequence.

- **ID:** BPMNChor11
- **Name:** Event-based Gateway – Deadline
- **Figures:** See Figure 8, Figure 9, and Figure 10.
- **Description:** An answer only has a limited time frame to be received. Participant B receives a message from Participant A. Participant B has to answer within a given timeframe (N-Time) or else another workflow will be triggered. Participant A has the timing responsibility.
- **Rules:** Participant B has to initiate the answering message. A Two-Way communication is required.

- **Used BPMN Elements:** startEvent (none), messageStartEvent, participant (pool), Message, originating from the initiator, messageStartEvent, timerStartEvent, endEvent (none).
- **Used Patterns:** This pattern is based upon the *Sequence Flow – Two Participants* pattern (to be published) with the restriction that the receiving participant has to answer in the given timeframe.
- **Synonyms:** Asynchronous Request-Response
- **Variations:** None.
- **Typical combinations:** This pattern may be inserted into any request-response workflow when a timing-based component is needed.
- **Use-Case:** Setting a Deadline for paying an invoice. If the time is over, a reminder may be sent. For a longer scenario, see Section VIII.

E. Open Parallel Gateway – Different Senders

The *Open Parallel Gateway – Different Senders* divides the interaction into two paths, with different senders (initiators). The order of the tasks in the respective paths can take place in parallel, they have no fixed order.

- **ID:** BPMNChor09
- **Name:** Open-Parallel Gateway – Different Senders
- **Figures:** See Figures 11, 12, 13.
- **Description:** The parallel gateway with different senders visualizes a parallel message dispatch of the participants activated in the task before the gateway.
- **Rules:** Participant B has to initiate the answering message. A Two-Way communication is required.
- **Used BPMN Elements:** startEvent (none), messageStartEvent, participant (pool), Message, originating from the initiator, messageStartEvent, timerStartEvent, endEvent (none).
- **Used Patterns:** None.
- **Synonyms:** Fire-and-Forget, Publish/Subscribe
- **Variations:** There can also be responses from the receivers, since the gateway is not associated with any participant, it is never closed.
- **Typical combinations:** Basic patterns can be used within the gateway.
- **Use-Case:** Send payment request and send policy, see Section VIII.

VI. Real User Connectivity

Another challenge in choreography is to implement communication with a real user, such as a staff member or a customer. In an orchestrated environment (e.g., using Camunda [10]), the orchestrator would take over the management of communication and also the tasks for the respective user (called tasklist).

Since this coordinator does not exist in the choreography, we have decided to abstract the user via an agent service. Accordingly, another service is added to realize the communication with the user. We are currently planning to implement

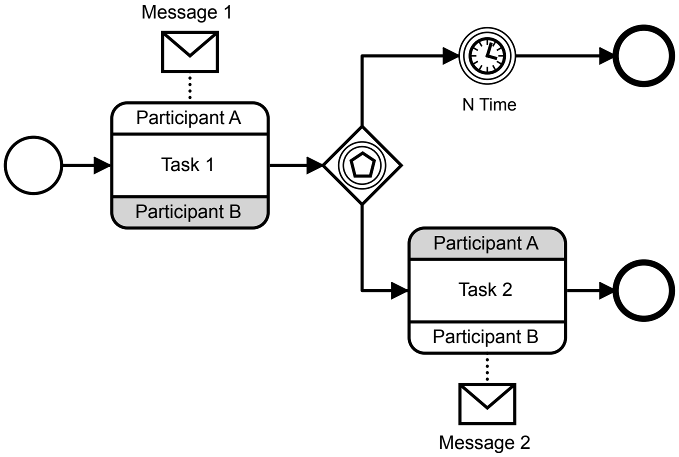


Figure 8. Event-based Gateway – Deadline Choreography.

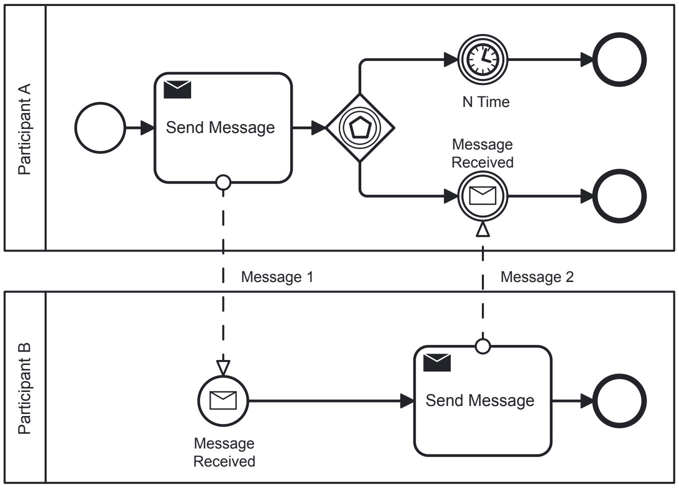


Figure 9. Event-based Gateway – Deadline Collaboration.

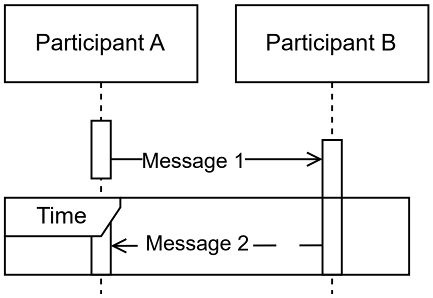


Figure 10. Event-based Gateway – Deadline UML Sequence.

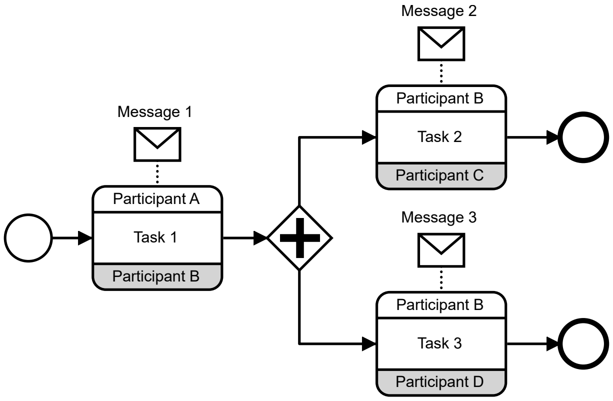


Figure 11. Open-Parallel Gateway – Different Senders Choreography.

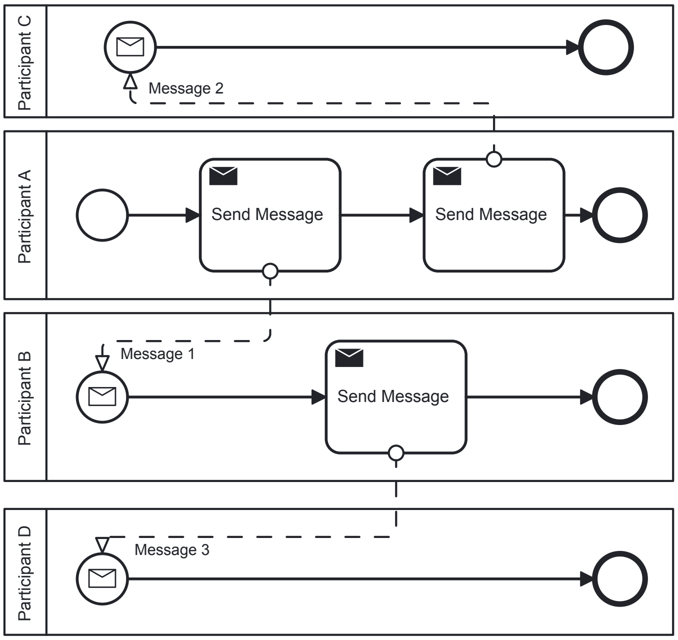


Figure 12. Open-Parallel Gateway – Different Senders Collaboration.

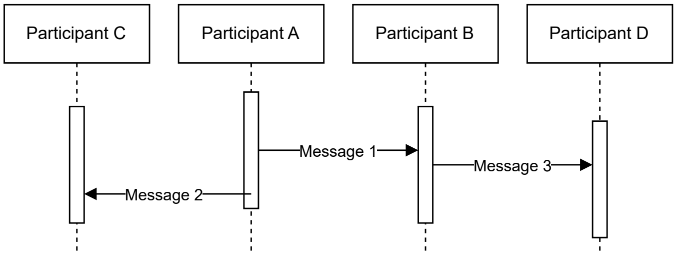


Figure 13. Open-Parallel Gateway – Different Senders UML Sequence.

this via a mail interface, but any other type of interface is conceivable in the future.

As an initial example, some information is sent to a customer (see Figure 14). The service represents an automated service within an MSA. In this case, the agent also simulates a task list for the user.

The actual implementation is visualized in Figure 15. In addition to the service and the customer, a customer agent is introduced that takes care of sending and receiving mail.

Sending mails is implemented in such a way that the agent service listens for events for the respective user, reacts to incoming events, generates and sends mails from these events. Receiving mails works the opposite way. As soon as a customer sends a mail to the agent service, the agent service converts the mail into an event and publishes it. Services within the MSA can then receive and process these events.

The agent service is not limited to an actual implementation type. In the first iteration, we decided to use a mail service to test the feasibility of this realization.

VII. CHOREOGRAPHY GRAMMAR

We are currently creating a grammar to specify the allowed interactions. This grammar validates the order of the patterns used to create valid interactions and prohibit invalid ones. However, some rules coming from the patterns, for example requirements for the participants, cannot be verified using the grammar without overcomplicating it. Therefore, additional constraints may be necessary.

The basic idea is that the letters of the grammar represent the patterns and BPMN elements so that a created word is an interaction to represent a choreography. Currently, it is a context-free language, since all relations R correspond to the form $N \times (N \cup T)^*$ [26]. The grammar contains all the patterns in the catalog.

Due to the complexity of the language and the rule set, only the non-terminals, terminals and the rules that describe the patterns from Section V are introduced here.

A. Notations $^+$ and *

The Kleene plus notation is used to extend the Kleene star notation for the grammar. This helps keep the derivation rules more concise [26].

- Using X^+ for at least once, or more
- Using X^* for arbitrary amount, or zero

B. ANY

ANY is a placeholder non-terminal for any other non-terminal of the pattern language, except start and end events. The use of *ANY* makes the grammar more concise.

It can be replaced by epsilon ε, i.e., an empty letter. Furthermore, there is no $ANY \Rightarrow ANY\ ANY$ within Rule 2, as precise rules are specified for each pattern as to when which patterns can come before or after.

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

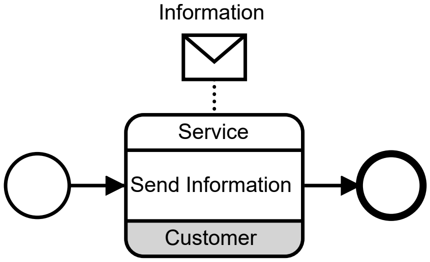


Figure 14. Abstract Example of Real User Connectivity — Choreography.

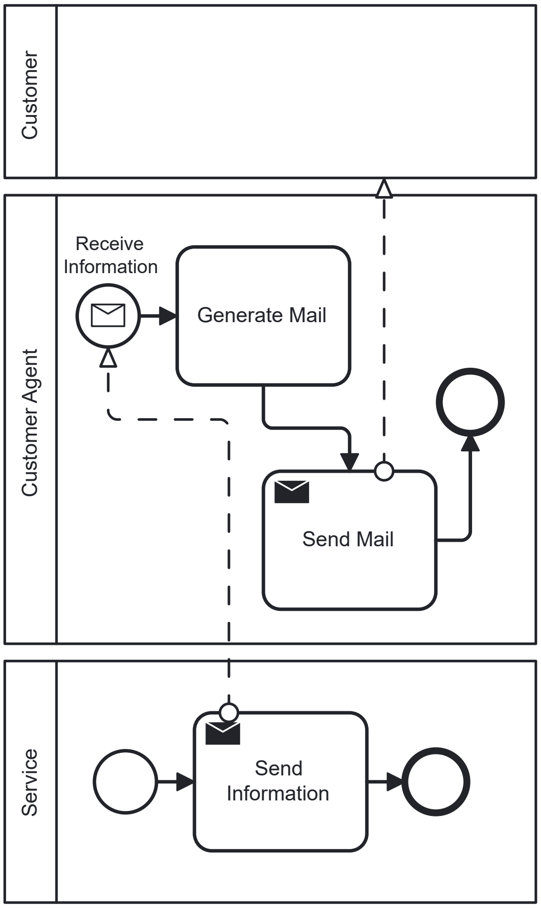


Figure 15. Abstract Example of Real User Connectivity — Collaboration.

C. Start and end event

Start and end are not necessary in a choreography. We force them into the grammar to make them more explicit.

- Non-Terminals: E_s for **s**tart event and E_e for **e**nd event
- Terminals: e_s for **s**tart event and e_e for **e**nd event
- Therefore, Rule 3: $E_s \Rightarrow e_s$, and Rule 4: $E_e \Rightarrow e_e$

D. Atomic patterns

The atomic patterns are described as follows:

- Described by One-Way Task and Two-Way Task.
- Non-Terminals: T_{ow} for **O**ne-**W**ay Task and T_{tw} for **T**wo-**W**ay Task
- Terminals: t_{ow} for **O**ne-Way Task and t_{tw} for **T**wo-Way Task
- Therefore, Rule 5: $T_{ow} \Rightarrow ANY\ t_{ow}\ ANY$ and Rule 6: $T_{tw} \Rightarrow ANY\ t_{tw}\ ANY$

E. Gateway patterns

To differentiate the gateways and also the type of open or closed, there are several non-terminals and terminals for creating a gateway and one universal terminal for closing, i.e., g_{close}. This terminal defines whether paths are merged again. As we have only used open gateways in our research work so far, we will not go into this implementation in more detail now, but in future work.

1) Event-based Gateway - Deadline: The Event-based Gateway — Deadline pattern has the following structure:

- Non-Terminal: G_{rd} for **r**ace-condition and **d**eadline
- Terminal: g_{rd}
- Using Rule 7: $G_{rd} \Rightarrow ANY\ (t_{ow}|t_{tw})^+\ ((g_{rd}[ANY\ E_e])\ (g_{rd}[e_{it}\ ANY\ E_e]))$
- where $ANY\ (t_{ow}|t_{tw})^+$ is an exchange starting the race condition,
- and $(g_{rd}[ANY\ E_e])$ as the path that can/should be executed within the defined time,
- and $(g_{rd}[e_{it}\ ANY\ E_e])$ as a path that describes the timer (i.e., e_{it}) and executes something and/or ends the process after expiry.

2) (Open-)Parallel Gateway: Since there is no verification of senders and receivers within the choreography task, there is no differentiation in the grammar between the Open-Parallel Gateway — Same Senders and Different Senders patterns. In addition, the implementation using the terminal g_{close} means that there is no direct distinction between open and closed gateways at the time they are created. The following describes the realization of an open parallel gateway:

- Non-Terminal: G_p
- Terminal: g_p
- Using Rule 1: $C \Rightarrow E_s\ ANY\ G_p$
- and using Rule 8: $G_p \Rightarrow ANY\ (t_{ow}|t_{tw})^+\ (g_p[ANY\ E_e])^+$
- where $ANY\ (t_{ow}|t_{tw})^+$ as an exchange starting the parallel message flow,
- and $(g_p[ANY\ E_e])^+$ as the definition of the parallel paths.

F. Derivation Rules

The derivation rules for the grammar are defined as follows: $R = \{$

1) $C \Rightarrow E_s\ ANY\ (G_x|G_p)$,
2) $ANY \Rightarrow T_{ow}|T_{tw}|G_p|G_{rd}|\varepsilon$,
3) $E_s \Rightarrow e_s$,
4) $E_e \Rightarrow e_e$,
5) $T_{ow} \Rightarrow ANY\ t_{ow}\ ANY$,
6) $T_{tw} \Rightarrow ANY\ t_{tw}\ ANY$,

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

7) $G_{rd} \Rightarrow$
$ANY\ (t_{ow}|t_{tw})^+\ ((g_{rd}[ANY\ E_e])\ (g_{rd}[e_{it}\ ANY\ E_e]))$,
8) $G_p \Rightarrow$
$ANY\ (t_{ow}|t_{tw})^+\ (g_p[ANY\ E_e])^+$,

}

G. Structure of each interaction

Each choreography follows the same structure: C is the start symbol of the choreography. The default structure is: $C \Rightarrow E_s\ ANY\ E_e$. Alternatively, the structure may be as follows: $C \Rightarrow E_s\ ANY\ (G_x|G_p)$ because in the case of open gateways (e**x**clusive or **p**arallel) there exists more than one end event.

Therefore, the choreography grammar (*G*) is currently described by $G = (N, T, R, S)$ with:

- $N = \{ANY, E_s, E_e, T_{ow}, T_{tw}, G_{rd}, G_p\}$ as Non-terminals, like a choreography, any patterns, or a specific pattern defined by a non-terminal shortcut,
- $T = \{e_s, e_e, e_{it}, t_{ow}, t_{tw}, g_{rd}, g_{gp}\}$ as terminals, for a specific pattern defined by a terminal shortcut,
- *R* as rules for the derivation from non-terminal to terminal, which are defined in Section VII-F,
- $S = C$, as a starting form of a choreography

The grammar is still in progress, so the content may change. After the presentation of the theoretical concepts, we will look at practical application in the next section.

VIII. PATTERN SCENARIOS IN INSURANCE COMPANIES

To realize the pattern language of the two introduced patterns in Section V completely, this section evaluates use cases of the patterns from the insurance industry and shows the implementation of the user and grammar.

We consider a typical process where a new insurance application is managed. The process *New Insurance Application* adopted from Freund and Rücker [17], but can also be taken directly from the insurance business model of our partners in the insurance industry, thus mapping a real-world use case. Due to the size of the process, it is only briefly described below and the parts containing the patterns are further explained.

A. Process and patterns

In the process, a customer submits a new insurance application. If the request is rejected, this information is noted in the backend and the customer is informed. If the request is accepted, a policy is created. After creation, the policy is sent and the customer is requested to submit the first payment. If the payment is not made within 60 days, the request, and the policy are invalid. If the customer pays in time, the insurance is valid.

The process starts with a synchronous communication using *Two-Way Task* pattern. The Application Service and Policy Service are then informed about the status of the interaction and the process splits into two parallel (or two independent) paths, implemented by the *Parallel Gateway — Different Senders* pattern.

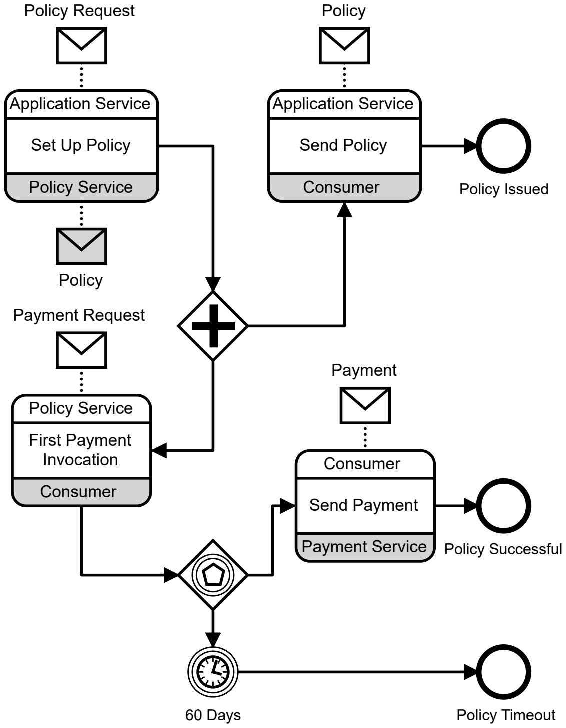


Figure 16. New Insurance Application Process – Cutout.

In one path, the *One-Way Task* pattern is represented, by sending the policy to the customer. In the other path, the *Event-based Gateway – Deadline* pattern is utilized by the sending and receiving of the payment request.

The process excerpt describes the happy path (see Figure 16), i.e., the customer obtains an insurance policy. It starts with the creation of the policy by the Application Service submitting a request to the Policy Service. The Policy Service responds by sending the policy. This interaction is realized using *Two-Way Task* pattern.

The interaction is then divided into two independent paths using the *Parallel Gateway — Different Senders* pattern, which means that the sequence of tasks in the respective paths can take place in parallel, their order is not fixed.

In the right path, the *One-Way Task* pattern is implemented. The application service sends the policy to the client. After sending, the task is completed and the path ends.

The *Event-based Gateway – Deadline* pattern is shown in the bottom path. The policy service sends the first payment request to the client. Then a timer is started. If the customer pays within 60 days, the policy, and the process are successful. If the customer does not pay within 60 days, a timeout occurs and the policy becomes invalid.

As shown with the payment request and the incoming payment in Figure 16, the *Event-based Gateway — Deadline* pattern contains the *One-Way Task* pattern. It shows that this

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

fundamental pattern is the basis of the minimal communication for the choreography.

B. User Connectivity

This subprocess also involves communication with a user, in this case the consumer. The Policy Service, which is an automated service within the MSA, sends the consumer a payment request. The consumer has to respond to this payment request within 60 days by paying the first insurance premium. Otherwise, the insurance status will expire.

The corresponding task in the choreography diagram is within the *Event-based Gateway – Deadline* pattern in the lower path of the *Parallel Gateway — Different Senders* pattern. This example shows the two paths, communication to the outside world and from the outside world to the system. The conversion to the outside world is described in Figure 17. The payment service generates a message, which the agent service converts into an email and sends to the consumer.

The implementation from the outside world is described in Figure 18. The consumer pays the first fee and an email is sent to the system, which is received by the agent service. The agent service converts the mail into a message and sends it to the policy service.

C. Grammar

Section VII described the grammar limited to the patterns presented here. Now the implementation is presented using a scenario. As the subprocess is only an extract, we represent the flow into the task *Set Up Policy* as a start event.

1) Start with the Rule 1: $C \Rightarrow E_s\ ANY\ G_p$
2) Derive $E_s \Rightarrow e_s$ using Rule 3 and $ANY \Rightarrow \varepsilon$ using Rule 4, so that $C \Rightarrow e_s\ G_p$
3) Derive $G_p \Rightarrow ANY\ (t_{ow}|t_{tw})^+\ (g_p[ANY\ E_e])^+$ using Rule 8,
   so that $C \Rightarrow e_s\ ANY\ (t_{ow}|t_{tw})^+\ (g_p[ANY\ E_e])^+$
4) Select a Two-Way Task in front of the gateway and set ANY to ε using Rule 4 after the start, so that
   $C \Rightarrow e_s\ t_{tw}\ (g_p[ANY\ E_e])^+$
5) Create two paths, so that
   $C \Rightarrow e_s\ t_{tw}\ (g_p[ANY\ E_e])(g_p[ANY\ E_e])$
6) Replace ANY in the first path using Rule 3 with a One-Way Task (*Send Policy* Task) and $E_e \Rightarrow e_e$ using Rule 4, so that
   $C \Rightarrow e_s\ t_{tw}\ (g_p[T_{ow}\ e_e])(g_p[ANY\ E_e])$
7) Replace One-Way Task Non-Terminal with its Terminal using Rule 5, so that $C \Rightarrow e_s\ t_{tw}\ (g_p[t_{ow}\ e_e])(g_p[ANY\ E_e])$
8) Replace ANY in the second path with an Event-based Gateway — Deadline (*Payment Procedure* Task) using Rule 2, so that
   $C \Rightarrow e_s\ t_{tw}\ (g_p[t_{ow}\ e_e])(g_p[G_{rd}\ E_e])$
9) Derive G_{rd} using Rule 7, so that $C \Rightarrow e_s\ t_{tw}\ (g_p[t_{ow}\ e_e])\ (g_p[ANY\ (t_{ow}|t_{tw})^+\ ((g_{rd}[ANY\ E_e])\ (g_{rd}[e_{it}\ ANY\ E_e]))])$

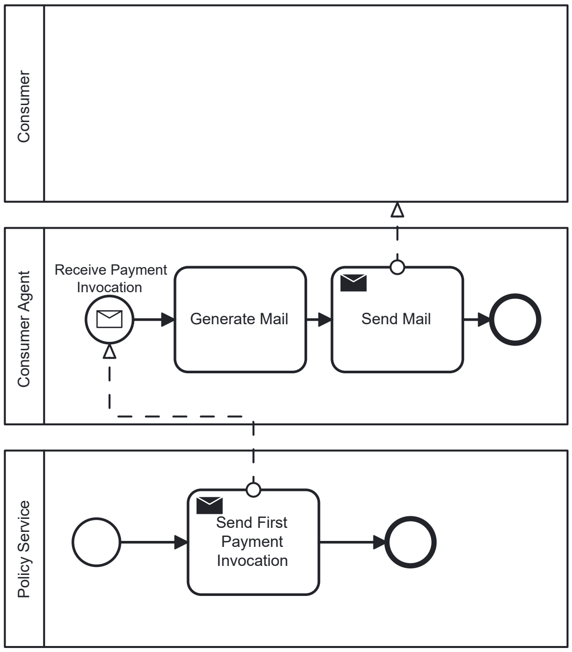


Figure 17. Realization User connection — to the outside world.

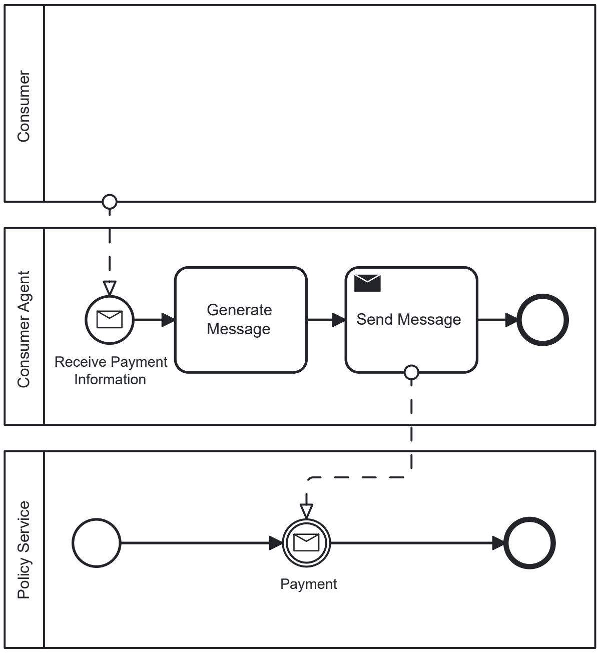


Figure 18. Realization of User connection - from the outside world.

10) Replace ANY before the first path of the race condition with ε using Rule 2 and select One-Way Task (*First Payment Invocation* Task), so that $C \Rightarrow e_s\ t_{tw}\ (g_p[t_{ow}\ e_e])(g_p[(t_{ow}\ ((g_{rd}[ANY\ E_e])\ (g_{rd}[e_{it}\ ANY\ E_e]))])$
11) Replace ANY in the first path of the race condition with a One-Way Task (*Send Payment* Task) using Rule 2, so that $C \Rightarrow e_s\ t_{tw}\ (g_p[t_{ow}\ e_e])(g_p[(t_{ow}\ ((g_{rd}[T_{tw}\ E_e])\ (g_{rd}[e_{it}\ ANY\ E_e]))])$
12) Derive the first path of the race-condition to the ter-

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

minals using Rule 6 and Rule 4, so that $C \Rightarrow e_s\ t_{tw}\ (g_p[t_{ow}\ e_e])(g_p[(t_{ow}\ ((g_{rd}[t_{tw}\ e_e])\ (g_{rd}[e_{it}\ ANY\ E_e]))])$

13) Derive ANY in the second path with ε using Rule 2 and replace the non-terminal from the end event using Rule 4 so that $C \Rightarrow e_s\ t_{tw}\ (g_p[t_{ow}\ e_e])(g_p[(t_{ow}\ ((g_{rd}[t_{tw}\ e_e])\ (g_{rd}[e_{it}\ e_e]))])$

The expression $C \Rightarrow e_s\ t_{tw}\ (g_p[t_{ow}\ e_e])(g_p[(t_{ow}\ ((g_{rd}[t_{tw}\ e_e])\ (g_{rd}[e_{it}\ e_e]))])$ describes the process as shown in Figure 16.

IX. CONCLUSION AND FUTURE WORK

The effective modeling and implementation of business processes is of crucial importance for an insurance company. Coming from BPMN notation, there needs to be a concise way of realizing the modeled process in the MSA style using the choreography. In this article, we further solidified the beginning of our choreography pattern language as the first steps towards a clear realization approach with precise implementation rules to map from BPMN diagrams to the distribution of microservices.

A way of integrating real users has been presented, to handle user interaction and allows the abstraction of these interactions. The grammar can validate the order of pattern usage, but cannot validate the rules of each individual pattern, an extra step for rule validation may be needed.

Several more patterns are needed to cover a broader range of different business use cases in the insurance industry. We also plan to evaluate all theoretical patterns with our insurance industry partners to ensure practical use. Additionally, an extension of the XML schema of BPMN diagrams is needed to serialize the patterns and validate them for correct usage. In future work, we will thus present additional patterns and an extension to the BPMN XML schema. We will also aim to refine our choreography pattern language and evaluate its additional benefit through a concrete implementation.

REFERENCES

[1] C. Schulze, A. Link, H. Meyer, A. Koschel, and A. Hausotter, “Towards Patterns for Choreography of Microservices-based Insurance Processes,” in *SERVICE COMPUTATION 2023, 15th Intl. Conf. on Advanced Service Computing*. IARIA, ThinkMind, 2023, pp. 1–7, Online. Available: https://www.thinkmind.org/index.php?view=article&articleid=service_computation_2023_1_10_10003 [retrieved: 02, 2024].

[2] Gesamtverband der Deutschen Versicherungswirtschaft e.V. - General Association o.t. German Insurance Industry, “VAA Final Edition. Das Fachliche Komponentenmodell (VAA Final Edition. The Functional Component Model),” 2001.

[3] European GDPR, “Complete guide to GDPR compliance,” Online. Available: https://gdpr.eu/ [retrieved: 02, 2024].

[4] Bundesanstalt für Finanzdienstleistungsaufsicht (BaFin) - Federal Financial Supervisory (BaFin), “Versicherungsaufsichtliche Anforderungen an die IT (VAIT) (Insurance Supervisory Requirements for IT (VAIT)) vom 03.03.2023,” 2023, Online. Available: https://www.bafin.de/DE/Startseite/startseite_node.html [retrieved: 02, 2024].

[5] C. Richardson, *Microservices Patterns: With examples in Java*. Shelter Island, New York: Manning Publications, 2018.

[6] S. Newman, *Building microservices: designing fine-grained systems*. Sebastopol, California: O’Reilly Media, Inc., 2015.

[7] C. Fehling, F. Leymann, R. Retter, W. Schupeck, and P. Arbitter, *Cloud Computing Patterns Fundamentals to Design, Build, and Manage Cloud Applications*. Springer Vienna, 2014.

[8] A. Koschel, A. Hausotter, R. Buchta, A. Grunewald, M. Lange, and P. Niemann, “Towards a Microservice Reference Architecture for Insurance Companies,” in *SERVICE COMPUTATION 2021, 13th Intl. Conf. on Advanced Service Computing*. IARIA, ThinkMind, 2021, pp. 5–9, Online. Available: https://www.thinkmind.org/articles/service_computation_2021_1_20_10002.pdf [retrieved: 02, 2024].

[9] A. Hausotter, A. Koschel, M. Zuch, J. Busch, and J. Seewald, “Components for a SOA with ESB, BPM, and BRM – Decision Framework and architectural Details,” *Intl. Journal od Advances in Intelligent Systems*, vol. 9, no. 3 & 4, pp. 287–297, 2016.

[10] “Workflow and decision automation platform,” Nov 2021, Online. Available: https://camunda.com/ [retrieved: 02, 2024].

[11] A. Koschel, A. Hausotter, R. Buchta, C. Schulze, P. Niemann, and C. Rust, “Towards the Implementation of Workflows in a Microservices Architecture for Insurance Companies – The Coexistence of Orchestration and Choreography,” in *SERVICE COMPUTATION 2023, 14th Intl. Conf. on Advanced Service Computing*. IARIA, ThinkMind, 2023, pp. 1–5, Online. Available: https://www.thinkmind.org/index.php?view=article&articleid=service_computation_2023_1_10_10002 [retrieved: 02, 2024].

[12] OMG, *Business Process Model and Notation (BPMN), Version 2.0*, Object Management Group Std., Rev. 2.0, January 2011, Online. Available: http://www.omg.org/spec/BPMN/2.0 [retrieved: 02, 2024].

[13] M. Fowler and J. Lewis, “Microservices a definition of this new architectural term,” 2014, Online. Available: https://martinfowler.com/articles/microservices.html [retrieved: 02, 2024].

[14] L. Krause, *Microservices: Patterns and Applications: Designing fine-grained services by applying patterns*. Lucas Krause, 2015.

[15] T. Allweyer, *Kollaborationen,Choreographien und Konversationen in BPMN 2.0 - Erweiterte Konzepte zur Modellierung übergreifender Geschäftsprozesse - Collaborations,Choreographies and Conversations in BPMN 2.0 - Advanced Concepts for Modeling Comprehensive Business Processes*. Fachhochschule Kaiserslautern, 2009.

[16] T. Allweyer, *Geschäftsprozessmanagement: Strategie, Entwurf, Implementierung, Controlling. - Business process management: strategy, design, implementation, controlling.* W3l GmbH, 2005.

[17] B. Rücker and J. Freund, *Praxishandbuch BPMN 2.0 - Practice Handbook BPMN 2.0*. Carl Hanser Verlag München Wien, 2014.

[18] G. Decker, O. Kopp, and A. Barros, *An Introduction to Service Choreographies*, vol. 50, no 2 ed. Information Technology, 2008.

[19] B. Rücker, “The Microservices Workflow Automation Cheat Sheet,” 2018, Online. Available: https://blog.bernd-ruecker.com/the-microservice-workflow-automation-cheat-sheet-fc0a80dc25aa[retrieved: 02, 2024].

[20] D. Mikalkinas, *Situation-aware Modelling and Execution of Choreography*. Stuttgart University, 2015.

[21] M. Milanović and D. Gasević, “Modeling service choreographies with rule-enhanced business processes,” in *2010 14th IEEE International Enterprise Distributed Object Computing Conference*, 2010, pp. 194–203.

[22] J. Ortiz, V. Torres, and P. Valderas, “A catalogue of adaptation rules to support local changes in microservice compositions implemented as choreographies of bpmn fragments,” 2023, Online. Available:https://riunet.upv.es/bitstream/handle/10251/181551/CatalogueOfAdaptationRules.pdf?sequence=1 [retrieved: 02, 2024].

[23] C. Richardson, “Eventuate Framework,” 2021, Online. Available: https://eventuate.io/[retrieved: 02, 2024].

[24] ——, “Eventuate Tram,” 2021, Online. Available: https://eventuate.io/abouteventuatetram.html[retrieved: 02, 2024].

[25] ——, “Eventuate Local,” 2023, Online. Available: https://github.com/eventuate-local/eventuate-local[retrieved: 02, 2024].

[26] G. Vossen and K.-U. Witt, *Grundkurs Theoretische Informatik (Basic Theoretical Computer Science)*, 6th ed. Wiesbaden: Springer Vieweg, 2016, publication status: Published.

[27] L. Priese and K. Erk, *Theoreotische Informatik - Eine umfassende Einführung (Theoretical computer science - A comprehensive introduction)*, 4th ed. Berlin, Heidelberg: Springer Vieweg, 2018, publication status: Published.

[28] J. Ladleif, A. von Weltzien, and M. Weske, “chor-js: A modeling framework for bpmn 2.0 choreography diagrams,” 2019.

[29] A. Barros, M. Dumas, and H. A.H.M, "Service interaction patterns," 2005, pp. 302–318, Online. Available:http://www.workflowpatterns.com/documentation/documents/serviceinteraction_BPM05.pdf [retrieved: 02, 2024].
[30] C. Chen, "Choreography vs orchestration," Online. Available: https://medium.com/ingeniouslysimple/choreography-vs-orchestration-a6f21cfaccae [retrieved: 02, 2024].
[31] B. Rücker, "The Microservices Workflow Automation Cheat Sheet," Online. Available: https://blog.bernd-ruecker.com/the-microservice-workflow-automation-cheat-sheet-fc0a80dc25aa [retrieved: 02, 2024].
[32] ——, *Practical Process Automation - Orchestration and Integration in Microservices and Cloud Native Architectures*. O'Reilly, 2021.

Digital Wallets and Identity Management: Pioneering Advances for Cloud Service Evolution

Fatemeh Stodt and Christoph Reich
Institute for Data Science, Cloud Computing, and IT Security; Furtwangen University,
Robert-Gerwig-Platz 1, 78120 Furtwangen, Germany
{Fatemeh.Stodt, Christoph.Reich}@hs-furtwangen.de

Abstract*—In today's technology-driven world, the management of digital identities has become a crucial concern. This is mainly because of the widespread use of online services and digital devices. The widespread use of digital platforms has created a complex web of online identities, placing the responsibility of juggling numerous usernames, passwords, and authentication methods on individuals. Digital wallets have emerged as a promising solution to tackle this complex challenge. This text highlights the versatility of these tools, which allow users to securely store, efficiently manage, and effectively utilize their digital assets, including personal data, payment information, and various credentials. In addition, the field of digital identity management has seen the rise of federated services, which provide users with the convenience of accessing multiple services using just one digital identity. An exceptional example in this field is Gaia-X, an innovative initiative focused on creating a reliable and secure data infrastructure. Gaia-X showcases the immense potential of federated services in bolstering digital identity management. This paper delves into a comprehensive examination of digital identity management, specifically examining the use of digital wallets and federated services. Our investigation delves into the categorization of identities needed to access various cloud services, taking into account their distinct requirements and characteristics. In addition, we explore the ever-changing world of digital wallets and federated identity management in the cloud. This sheds light on the upcoming requirements, challenges, and advantages. In addition, we present a thorough categorization scheme for cloud services, distinguishing them based on their security and privacy requirements. In this framework, we demonstrate the strategic mapping of different identity types to each category, providing a practical approach to aligning identity measures with the specific services being accessed.

Keywords-Digital wallet; Identity management; Federated service; Cloud.

I. INTRODUCTION

The management of digital identities has become a crucial issue in our modern digital era [1]. In today's digital landscape, the constant use of online services and the integration of digital devices into our daily routines have led to a situation where individuals are constantly sharing personal information to access a wide range of digital platforms, services, and applications [2]. The constant demand for digital identities has created a complex and fragmented landscape, where users are left with the task of managing multiple usernames, passwords, and authentication methods [3]. This situation, although aimed at guaranteeing access and security, frequently presents itself as a tangled web of complexity and vulnerability.

In this context, personal identity is spread across various digital domains, each requiring a unique set of keys for access. The fragmentation of identity and the proliferation of digital keys highlight the complex nature and risks of our interconnected digital world [4]. Managing multiple digital personas can be a cognitive challenge for users. Each persona is safeguarded by a unique combination of characters and security measures. This practice, although necessary, can be mentally taxing and increases the chances of security breaches. The more passwords one has to handle, the greater the risk of encountering vulnerabilities in the security system.

Clearly, the digital identity landscape has become a challenging arena where convenience and security often clash. The task at hand is to create creative solutions that not only simplify the management of digital identities but also strengthen their ability to withstand the constantly changing threats of the digital world [5]. This paper delves into the realm of digital identity management, seeking to clarify its complexities, address its challenges, and propose potential solutions in a world that is becoming more interconnected and data-driven.

Amidst these challenges, digital wallets have emerged as a promising solution for efficient digital identity management [6]. These software applications provide users with a convenient way to store, manage, and deploy a wide range of digital assets, including personal data, payment information, and various credentials [7].

In addition, the groundbreaking Gaia-X initiative exemplifies the importance of federated services and the considerable advantages they provide (Braud et al., 2021). Gaia-X has been meticulously designed to provide users with a strong and reliable data infrastructure, giving them unparalleled control over their personal information [8]. With Gaia-X's unified digital wallet, users can effortlessly navigate various platforms and services using just one digital identity. This simplifies the management of their digital presence and enhances security and privacy. This approach prioritizes the needs of users, giving them more control and encouraging innovation and competition in the digital realm. It strengthens the benefits of Gaia-X federated service model.

In today's digital landscape, the significance of efficient digital identity management cannot be emphasized enough. Luckily, there are some promising solutions on the horizon to address this complex challenge. Digital wallets and federated services are emerging as potential avenues to explore [9]. This paper aims to provide a thorough exploration of digital identity management and shed light on its practical uses, specifically in the areas of digital wallets and federated services. In addition,

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

we will delve into the use of digital wallets for accessing cloud services, offering valuable insights into their advantages and possible obstacles.

Here is the structure of this paper: Section II provides important background information on digital wallets and federated services. In Section III, we explore the requirements for identity management within wallets, specifically in the context of cloud access. In Section IV, we explored the benefits of digital wallets for cloud-based identity management. Section V provides a comprehensive categorization of cloud access based on identity group levels, offering valuable insights into the complex nature of the process. Section VI provides a use case that helps to further understand our approach. In Section VII, we bring together our findings and present a comprehensive conclusion that captures the essence of our exploration.

II. BACKGROUND (STATE OF THE ART)

This section provides an overview of three key components: Digital Wallets, Federated Services, and Federated Identity Management System, which play pivotal roles in ensuring secure and efficient digital experiences.

A. *Digital Wallet*

The rapid acceleration of digitization in transactions has been greatly influenced by the global pandemic, leading to an increased dependence on electronic services. Users today are actively involved in a variety of activities, including tax declarations, accessing vaccination and test certificates, and interacting with public administrations. These tasks are primarily carried out through digital platforms [10]. Users must complete an authentication process to access these services, ensuring electronic identification (eID) and protecting personal information. The authentication processes rely on identity management (IdM) systems to serve as gatekeepers, ensuring reliable and secure user authentication [11].

In this ever-changing digital landscape, digital wallets have become crucial elements in the realm of digital identity management. A digital wallet functions as a secure and encrypted repository, allowing users to effectively store and manage their digital identities, credentials, and other relevant information [12]. This serves as a crucial hub, providing a secure haven for users to store a wide range of authentication data, including usernames, passwords, digital certificates, and more [13].

There are a multitude of advantages associated with digital wallets in the field of identity management, which are both extensive and persuasive [14]. One of the standout features is the incredible convenience it provides, allowing users to store all their identities in one place, regardless of the services and platforms they use. In the safe and convenient realm of their digital wallet, users can easily store and organize multiple sets of credentials, eliminating the need to remember separate usernames and passwords for each service provider. This efficient integration of identity management not only streamlines the user experience but also significantly reduces the mental strain associated with managing multiple identities [15].

Throughout the years, a multitude of models for identity management systems have been developed and put into action. The earliest and most common model is the isolated model, which states that each service provider operates its own identity provider (IdP) [16]. Unfortunately, this approach places a heavy burden on users, as they are required to register with each service provider individually. This can be quite overwhelming, as it means having to manage multiple sets of credentials. This challenge prompted the introduction of the central identity model, which revolutionized the way IdP functionality is handled. It involves outsourcing the IdP functionality to a central entity that can be utilized by multiple service providers collectively [17]. Under this model, users only need to register once with the central IdP. After that, they can easily access a wide range of services using the same credentials.

Although the central identity model undeniably improves usability, it also brings up valid concerns about the central IdP being a potential single point of failure and a target for privacy breaches. The federated IdM model was introduced as a visionary approach that forges trust relationships among multiple IdPs, addressing these concerns [18]. Users registered with one IdP can easily authenticate themselves to service providers served by other IdPs within a circle of trust in this model. An excellent example is the European eIDAS interoperability framework, which effectively coordinates cross-border authentication processes by connecting national IdM systems across EU Member States.

The user-centric IdM model takes a different approach, with identity data being stored within the user's domain. This data is typically found on a smartcard or a smartphone equipped with a hardware-based security element [19]. Users maintain a strong sense of control over their identity data in this model, resulting in heightened privacy. Notable instances of this model include national IdM solutions that utilize smartcards effectively, like the Austrian Citizen Card and the German eID. The authentication processes retrieve the necessary identity information from the user's domain and seamlessly relay it to the requesting service provider, which improves control and privacy.

Nevertheless, the ever-changing digital identity landscape has given rise to the concept of Self-Sovereign Identity (SSI). This paradigm empowers users with complete control over their credentials, as recent advancements have demonstrated [20][14]. SSI represents a significant shift away from the conventional dependence on central authorities. Instead, it utilizes distributed ledgers among multiple IdPs operating within a circle of trust to register new credentials. Initiatives like the European Self-Sovereign Identity Framework (ESSIF) and Veramo exemplify this progressive approach. These developments showcase a noticeable transition towards user-controlled identity data, a trend that has garnered the interest of policymakers, as highlighted by the European Commission's proposal for a new European Digital Identity.

The OpenWallet Foundation (OWF) has established itself

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

as a leading force in the realm of digital wallets, showcasing innovation and creating new opportunities in this dynamic landscape [21]. OWF, under the Linux Foundation Europe, is dedicated to promoting open-source software development that enables interoperability among a wide range of wallet applications [22]. These applications cover a wide range of use cases, including secure payments, identity verification, and the secure storage of validated credentials. The vision of OWF perfectly aligns with the direction of a digital era characterized by user empowerment and the importance of secure, user-centric digital identities.

B. Federated Services

The concept of a federated catalog is crucial in the field of identity management as it enables the seamless discovery and access to a wide range of services through a centralized repository [8]. In this cutting-edge framework, various catalog systems work together to share important information about the services they offer. This collaborative effort results in a comprehensive and user-friendly resource center, simplifying the sometimes complex process of finding services [23].

The inter-catalog synchronization is a crucial element at the core of this federated catalog model. This crucial aspect guarantees that information regarding services, including their availability, descriptions, and detailed attributes, is consistently kept up-to-date and aligned across various catalog systems [24]. The updates and modifications made within one catalog seamlessly reverberate across others, thanks to the intricate mechanisms of inter-catalog synchronization. This carefully executed coordination not only maintains the integrity of the data but also ensures that users receive precise and up-to-date information. The result is a smooth user experience where people can trust the federated catalog to provide consistent and unified information about the services available [25].

The incorporation of digital wallets into the structure of federated catalogs brings an extra level of functionality and convenience to the field of identity management [25]. The seamless integration of digital wallets and federated catalogs enhances the efficiency of service discovery and access, as they securely store and manage users' digital identities and associated credentials. When a user interacts with the federated catalog using their digital wallet, the wallet plays a vital role in verifying the user's identity and sharing relevant identity information with the catalog. The symbiotic interaction allows the catalog to provide personalized service recommendations, curate search results to meet individual needs, and smoothly manage authentication and authorization processes [26]. The end outcome is an enhanced user experience, carefully adjusted for both user convenience and security.

It's worth mentioning that federated services and federated catalogs are closely connected concepts in the field of identity management. Federated services rely heavily on federated catalogs as crucial resources, necessary for providing users with a centralized and comprehensive view of the services available. Thanks to this collaboration, users now have the convenience of exploring and accessing services with just one digital identity. This powerful collaboration between service providers and catalogs, operating under the federated model, efficiently simplifies identity management processes. The catalog serves as a reliable intermediary, managing authentication, user authorization, and facilitating smooth information exchange between users and service providers [27]. The smooth functioning of the federated ecosystem relies heavily on the crucial role played by federated catalogs, which serve as linchpins in the complex task of managing identity.

C. Federated Identity Management System

The Federated Identity Management System (IdMS) marks a notable departure from the conventional centralized IdMS concept. The federated IdMS system connects user identity information across multiple organizations, as shown in Figure 1. This approach avoids relying on a single central identity provider. In this dynamic model, the service providers and identity providers are entities that represent trusted organizations within the federation.

At the heart of this framework lies the importance of fostering a collaborative agreement among the different service providers. This agreement guarantees that a user's identity is not just acknowledged but also universally accepted by all participating service providers within the federation. The seamless access to services from multiple providers within the federation allows users to avoid the hassle of repeated authentication or the need to create separate accounts for each service [28]. This collaborative approach improves user convenience and promotes a greater sense of trust within the federation.

The federated IdMS is a powerful and versatile solution that allows users to effortlessly access a wide variety of services, all while ensuring their identity remains secure and consistent across trusted organizations. This represents a notable departure from the conventional centralized identity model, providing improved adaptability, scalability, and user-focused identity management.

The federated IdMS model introduces the concept of a user needing to authenticate themselves only once to access multiple services, making it more convenient for users. This efficient process is known as Single Sign-On (SSO), and it relies on the established trust between service providers and the identity provider [28]. SSO operates by utilizing security tokens to enable secure and efficient access to a range of services.

III. REQUIREMENTS FOR IDENTITY WALLETS FOR FUTURE CLOUDS

In today's rapidly evolving digital landscape, the importance of robust identity management cannot be overstated. With the increasing prevalence of cloud computing, it is crucial to establish secure and seamless access to cloud services. This section explores the categorization of identities needed for various cloud services, examines their specific requirements and characteristics, investigates the future needs for digital wallets and federated identity management in the cloud, identifies possible challenges in implementing identity wallets for future

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

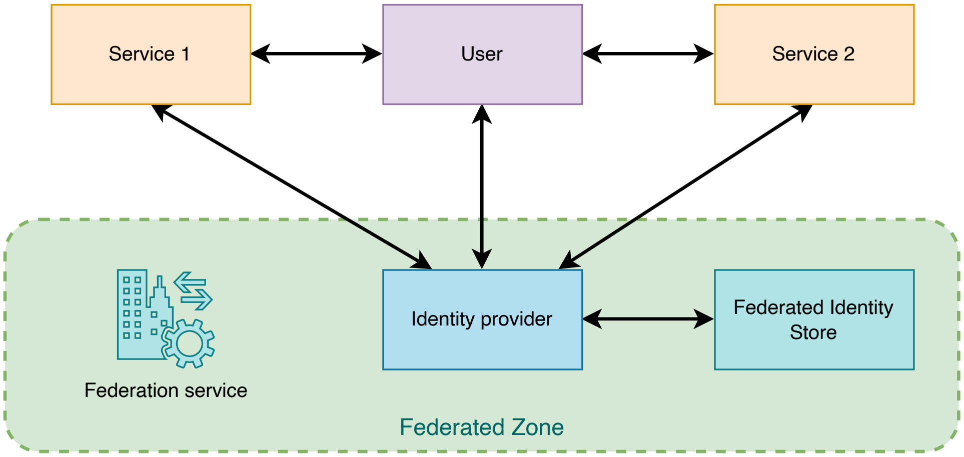


Figure 1. Dataflow Representation of a Federated Identity Management System

clouds, and emphasizes the potential advantages of using digital wallets for identity management in the cloud.

Understanding the diverse identity landscape becomes more comprehensive when identities are categorized based on their usage in various cloud services. The identities mentioned can be categorized into user identities, service identities, and device identities [29]. The different types of identities in the cloud ecosystem include user identities, service identities, and device identities. User identities are used by individuals to access cloud services, while service identities are specifically associated with cloud services or applications. Device identities, on the other hand, are used to authenticate and authorize devices that interact with cloud resources. Effective identity management requires careful consideration of the unique requirements and characteristics of each identity category.

R1: Secure storage of identity-related data: In the ever-changing landscape of cloud services, the protection of identity and identity-related data becomes a top priority. This essential requirement is based on the importance of maintaining the confidentiality, integrity, and availability of sensitive data associated with user identities, service identities, and device identities [13].

When it comes to cloud services and the movement of data across virtual boundaries, the importance of secure storage cannot be emphasized enough. It forms the foundation for establishing trust and ensuring data protection. This requirement is crucial in protecting the digital identities that are essential for accessing cloud resources.

Secure storage goes beyond just keeping data. It encompasses much more than that. It demonstrates a strong dedication to implementing thorough security measures that protect identity-related information. The implementation involves the use of strong cryptographic measures, strict access controls, and robust data encryption methods. The implementation requires the use of intrusion detection systems and ongoing monitoring to protect user identities, service credentials, and device authentication data from unauthorized access and potential harm. Through this approach, secure storage effectively minimizes the potential dangers linked to data breaches, identity theft, and unauthorized access.

R1 exemplifies the fundamental principle of trust in cloud-based identity management. This serves as a crucial safeguard, preserving the integrity of identity-related data and strengthening the foundation of cloud security and user trust.

R2: Effective management of identity-related data: Managing identity-related data within the cloud is a complex task that involves a range of important functionalities. The functionalities cover a wide range of tasks, from carefully curating data to thoroughly reviewing it, all done within the cloud environment. In addition, it is crucial to have the ability to determine which aspects of identity data should be shared beyond the cloud's boundaries [30]. R2 aims to give users a greater sense of control over their stored information, promoting privacy and enabling more nuanced data management.

R2 goes beyond basic data management and instead encourages a feeling of digital self-determination within the expansive realm of the cloud. This requires the ability to carefully select, modify, or delete portions of identity data stored in cloud repositories. Users have the power to shape their digital personas within the cloud ecosystem, making thoughtful decisions about which aspects of their identity should be accessible. This falls under the purview of R2.

The concept of selective data sharing is crucial to R2's mandate. This allows users to determine which aspects of their identity data should be shared with external entities or services, extending beyond the boundaries of the cloud. The careful management of data sharing provides a strong privacy mechanism, which enhances user trust in the cloud's identity management processes.

R2 serves as a protector of user autonomy, creating an environment where individuals have precise control over their digital identities. R2 prioritizes privacy and data sovereignty by incorporating these capabilities into the cloud environment. It highlights the idea that users play an active role in managing their identity-related information, rather than being passive recipients. By doing so, it strengthens the core principles of trust and user-centricity in cloud-based identity management. This ushers in a new era where users are not just data subjects, but rather thoughtful guardians of their digital identities.

R3: Secure sharing of identity-related data: The importance of R3 becomes evident in the complex realm of cloud-based identity management. This involves the facilitation of secure sharing of identity-related data beyond the boundaries of the cloud environment, which is of utmost significance. This undertaking relies on the creation of robust and secure communication channels and protocols, coordinating the distribution of identity data to trusted entities [31]. It is crucial to maintain unwavering commitment to upholding data integrity and confidentiality throughout the data-sharing process.

R3 encapsulates a fundamental dichotomy. It acknowledges the importance of sharing identity-related data for the smooth operation of digital services, applications, and interactions. However, it emphasizes the absolute need for strong protections to surround this data as it moves beyond the confines of the cloud.

This task involves the development of strong communication channels that can withstand potential adversarial forces. These channels serve as pathways for identity data to travel to its intended recipients. At the same time, strong protocols are implemented to regulate the journey of this data. The protocols serve as digital guardians, ensuring the integrity and confidentiality of every packet of information.

In addition, R3 highlights the intricate relationship between trust and technology. The cloud environment must foster a sense of trust, allowing users to securely entrust their identity data to the digital realm. The trust in our system is reinforced by a comprehensive set of technical measures that effectively protect against data breaches, eavesdropping, and tampering.

R3 effectively resolves the dilemma of sharing data while ensuring its protection. This statement highlights the importance of protecting identity-related data as a means to safeguard individual privacy and security in the digital realm. R3's commitment to secure sharing harnesses the power of the cloud to effectively balance the need for seamless service delivery and robust data protection. In the realm of cloud-based identity management, it serves as a strong foundation where trust and security work together to create a thriving digital ecosystem that is safe and reliable.

R4: Secure storage of cryptographic material: R4 is a crucial necessity in the constantly changing world of cloud-based identity management. This mandate emphasizes the importance of creating a secure vault to protect the cryptographic components of digital identities within the cloud ecosystem [32]. This requirement emphasizes the importance of protecting cryptographic elements, such as keys and certificates, from unauthorized access, ensuring their confidentiality and security.

R4 emphasizes the importance of cryptographic material as the foundation of trust in the digital world. The cryptographic artifacts, which include cryptographic keys and digital certificates, play a crucial role in verifying users, validating services, and ensuring the security of data exchanges. R4 has a dual purpose. It emphasizes the need for strong cryptographic storage mechanisms that can effectively protect against intrusion, manipulation, and compromise. These repositories ensure that cryptographic material is kept securely protected, safeguarded from any potential harm caused by malicious individuals.

Furthermore, R4 emphasizes the importance of maintaining confidentiality. The text emphasizes the importance of robust cryptographic measures to safeguard sensitive information. It highlights the need for multiple layers of protection to ensure the confidentiality of cryptographic secrets, even in the event of a breach. The maintenance of confidentiality relies heavily on cryptographic mechanisms like encryption, digital signatures, and access controls.

In addition, the imperative embedded within R4 goes beyond simple storage. The entire lifecycle of cryptographic material is covered, including generation, distribution, rotation, and retirement. Every aspect of this lifecycle requires careful planning, strict security measures, and ongoing supervision to ensure the material remains resilient against emerging threats.

R4 plays a crucial role in maintaining trust in the cloud's identity management infrastructure. This serves as a guardian, guaranteeing that the cryptographic keys and certificates, which are crucial for secure digital identities, remain completely secure and cannot be compromised. The cloud demonstrates its dedication to upholding the importance of digital interactions, protecting user identities, and supporting the fundamental principles of security and trust in the digital world by adhering to R4's mandate.

R5: Combining identity data before sharing: The complex world of cloud-based identity management is enriched by the presence of R5, which seamlessly aligns with the profound concept of selective disclosure. The core concept of R5 emphasizes the importance of users having the ability to carefully merge and curate their identity data, aligning it with their specific sharing needs [33]. This skillful capacity to combine and distribute pertinent aspects of identity is crucial for maintaining privacy and managing the sharing of identity-related data.

The mandate outlined in R5 is rooted in the fundamental principle of selective disclosure. In the complex world of cloud computing, it is important for users to have control over how they disclose their identity. This allows them to customize their disclosures to suit their specific needs and preferences. This thoughtful approach to data sharing allows individuals to take control of their digital identities, sharing only the necessary information for each interaction. This not only protects their privacy but also gives them greater control over their identity-related data.

Practically speaking, R5 acts as a protector of personal privacy, giving users the ability to have precise control over their identity data. The necessity for extensive data sharing is eliminated, as it enables a more precise and refined approach to identity facets. The users have the ability to carefully construct their identity by selecting the relevant attributes, credentials, or personal details to include, customizing their disclosure for each unique situation.

R5 aims to promote a more sophisticated approach to data

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

sharing, moving away from a simplistic binary perspective. Users have the ability to find a delicate equilibrium between sharing the right amount of information to foster meaningful interactions, while also protecting their privacy. This approach not only improves personal data protection, but also fosters a sense of trust in cloud-based identity management.

R5 acts as a diligent sentinel, safeguarding the delicate equilibrium between privacy and data exchange in the expansive domain of the cloud. It highlights that in the era of digital technology, the act of sharing data is not necessarily a binary choice. The cloud ecosystem promotes a robust sense of privacy, enabling individuals to retain mastery of their digital identities with proficiency.

Using digital wallets for identity management in the cloud provides a variety of advantages. Digital wallets improve user convenience by offering a centralized platform for managing identities across various cloud services. The security measures are enhanced by implementing secure authentication mechanisms, employing strong encryption for identity data, and implementing efficient access control. In addition, digital wallets provide users with the power to have control over their personal information and the option to selectively share it with trusted entities. The integration of digital wallets with federated identity management enhances the efficiency of identity management processes, facilitating easy access to cloud resources and promoting compatibility.

Table I provides a comparison of various digital wallets, with a specific focus on their Identity Management (IdM) capabilities and wallet requirements. This table is an excellent resource, offering a concise summary of the strengths and functionalities of different digital wallet solutions in terms of identity management.

The table examines the authentication methods used by each digital wallet. These mechanisms play a crucial role in maintaining secure identity validation within the cloud environment. By familiarizing themselves with these approaches, users can make well-informed decisions regarding which wallet is most suitable for their individual security requirements.

Encryption Techniques: Security is of utmost importance in IdM, and the table reveals the encryption techniques used by each digital wallet. The techniques serve as a strong defense for identity data, ensuring its protection against potential threats. Users are empowered with the knowledge to effectively prioritize the protection of their identity data.

Controlling access to identity attributes is crucial for maintaining privacy and security. The table provides a comprehensive explanation of the access control features found in each digital wallet, detailing how they effectively handle the distribution of user identity information. Users can adjust their privacy settings to match their personal preferences and needs.

The table emphasizes the level of user control offered by each digital wallet in terms of personal information, which is crucial in today's data-driven world where ownership and autonomy are highly valued. This text offers valuable insights into how users can carefully choose which identity data to share with trusted entities, allowing them to maintain their privacy and control over their data.

Table I provides readers with a comprehensive understanding of the intricate features and capabilities of each digital wallet. With this knowledge at hand, stakeholders can confidently make decisions and choose a digital wallet solution that perfectly matches their specific identity management requirements. This table is a valuable tool that provides insights into effective, secure, and user-centric identity management in the constantly changing world of cloud services.

IV. Unlocking the Advantages of Digital Wallets for Cloud-Based Identity Management

Digital wallets have a significant impact on identity management in the cloud, offering a wide range of benefits that transform the way we engage in digital interactions. The virtual guardians play a crucial role in strengthening the foundation of cloud-based identity management. This section explores the numerous benefits that users gain from cloud services and the complex nature of cloud security.

Enhanced User Convenience: Digital wallets have become exemplars of user-centricity, creating a haven of convenience within the vast realm of the cloud. They appear as centralized repositories, giving users the ability to seamlessly manage their digital identities across a variety of cloud services. Gone are the days of managing a collection of credentials, each resembling a digital puzzle piece, as they slowly fade into obscurity. Users are introduced to a realm where a streamlined interface enhances their experience with the digital world. This transformation not only simplifies the user experience but also reduces the mental effort required to handle multiple digital identities.

Fortified Security Infrastructure: Digital wallets serve as strong protectors of security in the cloud. The authentication mechanisms they implement set a high standard, providing strong protection against unauthorized access. Within these virtual strongholds, identity data is securely protected by layers of robust encryption. This strong armor guarantees that even when faced with possible breaches, the identity data remains a mystery, protecting its integrity and confidentiality. The diligent guardians of access control oversee the allocation of identity attributes, granting access only to those who possess the necessary permissions to unlock the wallet.

Empowerment and Data Sovereignty: The concept of empowerment lies at the core of digital wallets. Users have the power to become guardians of their personal information, choosing who they share it with and maintaining control over it. This shift in power represents a new era, where individuals are at the forefront of their digital identities. Users are no longer passive participants, but rather discerning custodians of their identity data. This transformation represents a significant milestone in the age of digital interactions, where the control of data is placed in the hands of those who are most affected by it.

Seamless Integration with Federated Identity Management: The integration of digital wallets with federated identity

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

TABLE I. Comparison between different digital wallet based on IdM and wallet requirements

Reference	IdM	Environment	R1	R2	R3	R4	R5
[20]	SSI	Local	✓	✓	✓	✓	✓
[25]	Federated	Local	✓	✓	✓	✓	✓
[13]	Centralized	Local	✓	✓	✓	✓	✗
[34]	as a Service	Remote	✓	✓	✓	✗	✓
[19]	User-centric	Local	✓	✓	✓	✓	✗
[17]	Centralized	Remote or Local	✓	✓	✓	✓	✗
[14]	SSI	Remote or Local	✓	✓	✓	✓	✓
kudra2022self , [21]	SSI	Remote or Local	✓	✓	✓	✓	✓

management is like a harmonious collaboration within the expansive realm of cloud computing. This seamless blending of two elements results in a beautifully orchestrated identity management system. The integration acts as a crucial component that simplifies identity management processes, bringing about a time of effortless access to cloud resources. Users navigate the digital realm with a unified digital identity, surpassing the obstacles posed by various service providers. In this realm of collaborative synergy, creativity blossoms, and healthy competition thrives, highlighting the significant benefits of a federated service model.

Overall, digital wallets play a crucial role in ensuring user-centricity, security, and interoperability in the vast realm of the cloud. They have a unique perspective on the user experience, prioritize security, give individuals control over their data, and promote the effectiveness of federated identity management. In this era of rapid technological advancements, digital wallets serve as essential tools that help users navigate the complex realm of cloud-based identity management. They play a crucial role in maintaining trust, security, and user empowerment as unwavering principles.

V. Access management and Categorising Identities for Cloud Services

The significance of security and privacy in the dynamic field of cloud computing cannot be emphasized enough. We present a rigorous categorization approach that groups cloud services according to their distinct security and privacy needs in order to efficiently traverse this difficult terrain. This scheme offers practical insights into choosing the most suitable identity type for each service, in addition to providing a framework for comprehending the heterogeneous environment of cloud services.

As Figure 2 illustrates, our classification approach distinguishes between three categories of cloud services: low-security services, moderate-security services, and high-security services. Every layer is carefully crafted to conform to the differing levels of privacy and security requirements that are specific to certain service kinds.

A. Low-Security Services

Cloud offerings classified as low-security services are typically used for non-sensitive data and require only minimal protection measures. These services generally consist of information that is readily available to the public and poses little risk if it becomes exposed. There are various sources where you can find information, such as websites, public repositories, and blogs.

1) Identity Needs: Within the domain of low-security cloud services, the authentication and access control techniques depend on simple user identities, usually consisting of usernames and passwords. Due to their widespread presence, these user identities provide convenient management. Within this particular scenario, the main objective is to develop a solid foundation for safeguarding data rather than implementing strict safeguards to ensure privacy. Although privacy is still a worry, the focus is on adopting fundamental security measures, such as ensuring data protection during transmission through protocols like HTTPS. Nevertheless, the implementation of strong encryption and multifactor authentication is sometimes considered superfluous in these situations. The main goal is to prioritize simplicity and user-friendliness, making it easier for a wide range of users to access and use this specific category of cloud services.

2) Mapping Identities to Categories: For services in this category, basic user identities like usernames and passwords may be enough for authentication and access control. These credentials offer sufficient security for services with low privacy concerns.

B. Moderate-Security Services

Moderate-security services are positioned in the middle of the range, dealing with fairly sensitive data that necessitates stronger protective measures compared to low-security services. These services handle data of moderate sensitivity, such as personal information, internal organizational papers, or consumer data with a certain degree of secrecy.

1) Identity Needs: When dealing with the needs of cloud services that have moderate security requirements, there are various factors linked to identification that need to be taken into account. Improving security is of utmost importance, requiring the implementation of more robust authentication methods like two-factor authentication (2FA), biometrics, or the integration of extra security measures. At the same time, there is increased focus on privacy concerns due to the fact that the data being managed may have legal or compliance ramifications. In order to address these concerns, implementing data anonymization and controlled access mechanisms are considered appropriate safeguards. In addition, security precautions in this context include encrypting data both when it is stored and when it is

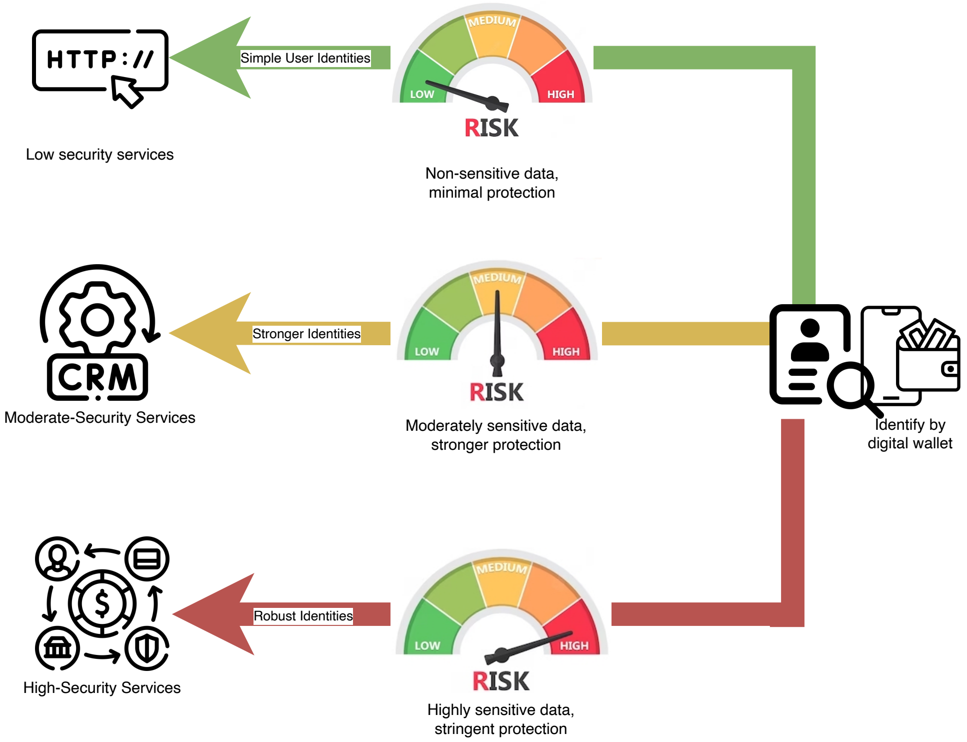


Figure 2. Categorisation of Cloud Services and Identity Types

being transferred. These steps are supported by regular security audits and checks to ensure compliance. In the middle of these efforts focused on security, there is a deliberate attempt to find a careful balance between improving security measures and maintaining a great user experience. This recognizes the significance of user convenience, especially in the context of cloud services with moderate levels of protection.

2) Mapping Identities to Categories: The moderate-security category requires more robust authentication procedures to enhance security. Implementing two-factor authentication or biometrics may be essential in order to ensure strong protection for personal information or organizational documents.

C. High-Security Services

High-security services are responsible for safeguarding extremely sensitive data and so must implement the most rigorous and demanding protective measures. These services manage highly confidential data, including bank records, health information, trade secrets, and classified information. Unauthorized access or breaches may result in significant repercussions.

1) Identity Needs: High-security services necessitate the implementation of the most rigorous protective measures. They employ multifactor authentication, which integrates various authentication techniques such as smart cards, biometrics, and one-time passwords to enhance identity verification. In addition, digital certificates are provided to enhance security by providing identification for both users and services. Hardware Security Modules (HSMs) are essential in safeguarding against advanced cyber threats since they offer tamper-resistant encryption and secure storage of cryptographic keys in hardware. Continuous surveillance and thorough examination of user behaviors also assist in the prompt identification of potential security breaches. In addition, the implementation of stringent access controls, such as role-based access control (RBAC) and the principle of least privilege, ensures that only authorized persons are granted access to sensitive information.

2) Mapping Identities to Categories: High-security services require the use of multifactor authentication, digital certificates, or physical tokens for the most sensitive data. By integrating these safeguards with rigorous surveillance and access restrictions, the utmost level of safeguarding and confidentiality is achieved, guaranteeing secure entry to vital information.

VI. PRACTICAL APPLICATION OF THE CATEGORIZATION SCHEME

Here, we demonstrate a tangible implementation of our classification system to showcase its practical use in real-life scenarios. In this case study, we illustrate how the scheme helps in identifying suitable security solutions for various cloud applications.

Case Study: Categorizing the Security of Cloud Services

Let's examine a company called CloudTech Inc. that provides a variety of cloud services. In order to gain a deeper understanding of the advantages of our categorization scheme, we will analyze three specific services it offers: a publicly accessible website, a Customer Relationship Management (CRM) system, and a Secure Financial Transactions (SFT) platform.

Service with minimal security measures - Website accessible to the general public:

CloudTech Inc. manages a website that is accessible to the public. The website offers comprehensive details about the company, its products, and valuable knowledge about the industry. This website solely functions as a promotional and informative tool and does not include the processing of sensitive data. As a result, it is classified as a low-security category in our system.

Identity Type: This service requires a basic user identity, which includes a username and password. Users necessitate fundamental authentication in order to have access to general information and resources on the website.

CloudTech Inc. employs industry-standard security measures, such as data encryption for transmission and safeguards against common online vulnerabilities, to guarantee the security of user interactions on the website.

CRM System with Moderate-Security Service:

CloudTech Inc. provides a cloud-based client Relationship Management (CRM) solution that effectively handles client data, such as contact information, purchase records, and conversation logs. This service entails the management of confidential client data and is therefore classified as having a moderate level of security.

The CRM system utilizes two-factor authentication (2FA) to efficiently protect the security and privacy of customer data. In addition, biometric authentication is used to increase security.

The CRM system employs strong encryption methods to protect data both while it is stored and when it is being transmitted. Access controls are used to guarantee that only authorized individuals have the ability to view or alter client records. The service is brought into alignment with industry standards and laws through regular security audits and compliance inspections.

Advanced Security Service - Secure Financial Transactions (SFT) Platform:

CloudTech Inc. provides a safe Financial Transactions (SFT) platform that enables safe monetary transactions, such as payment processing and cash transfers. This service handles extremely confidential financial information and consequently belongs to the high-security classification.

The SFT platform employs multifactor authentication (MFA), which includes biometrics, smart cards, and one-time passwords (OTP), to guarantee the utmost level of identity verification.

The security measures implemented for the SFT platform encompass various aspects. These include the utilization of end-to-end encryption to protect transaction data, conducting thorough penetration testing to identify vulnerabilities, continuous real-time monitoring of transactions, and adherence to strict financial sector rules.

Analysis of the Case Study:

This case study demonstrates the efficacy and flexibility of our categorization approach in real-life situations:

Customized Security: Our categorization method enables CloudTech Inc. to customize security measures according to the unique requirements of each service. Less secure services benefit from less complex authentication, whereas more secure services require the most rigorous identity verification methods.

CloudTech Inc. optimizes resource allocation by aligning security with service categories, resulting in efficient resource allocation. High-security protocols are specifically implemented for services that deal with confidential information, guaranteeing optimal utilization of security resources.

Data Protection and Privacy: The categorization scheme emphasizes the significance of safeguarding confidential information, advocating for compliance with legislation around data protection and privacy.

To summarize, our hierarchical classification system improves the decision-making process for adding security measures in cloud services. It guarantees that the level of security matches the level of sensitivity of data, enabling organizations such as CloudTech Inc. to offer strong protection for important information without introducing extra complications in less secure situations. This framework is a useful tool in the constantly changing field of cloud computing security.

VII. CONCLUSION

To summarize, digital wallets and federated services provide substantial benefits in the administration of digital identities. Digital wallets offer a secure and convenient method for users to store and oversee their digital assets, streamlining the management of digital identities while improving security and privacy. The development of various identity management methods, such as federated and user-centric approaches, alongside innovations like Self-Sovereign Identity (SSI), provide users more authority over their credentials. Initiatives such as Gaia-X demonstrate the objective of empowering consumers with greater authority over their personal data and promoting innovation in the digital domain.

In the future, it is important for research to concentrate on incorporating new technologies like blockchain and decentralized identification systems to improve the security and privacy of digital wallets and federated services. Furthermore, investigating the usability and user experience elements of these solutions might enhance their adoption and acceptance among users. Sustained endeavors in research and development will aid in tackling the intricate obstacles of digital identity management and guarantee its significance in the contemporary digital age.

ACKNOWLEDGEMENT

This research was funded by the Federal Ministry of Education and Research (BMBF) under reference number COSMIC-X 02J21D144, and supervised by Projektträger Karlsruhe (PTKA).

REFERENCES

[1] F. Stodt and C. Reich, “A review on digital wallets and federated service for future of cloud services identity management”, in *Service Computation 2023: The Fifteenth International Conference on Advanced Service Computing, June 26, 2023 to June 30, 2023, Nice, France*, 2023, pp. 16–20.

[2] A. Rashid and A. Chaturvedi, “Cloud computing characteristics and services: A brief review”, *International Journal of Computer Sciences and Engineering*, vol. 7, no. 2, pp. 421–426, 2019.

[3] S. Rajamanickam, S. Vollala, R. Amin, and N. Ramasubramanian, “Insider attack protection: Lightweight password-based authentication techniques using ecc”, *IEEE Systems Journal*, vol. 14, no. 2, pp. 1972–1983, 2019.

[4] G. Ra, T. Kim, and I. Lee, “Vaim: Verifiable anonymous identity management for human-centric security and privacy in the internet of things”, *IEEE Access*, vol. 9, pp. 75 945–75 960, 2021.

[5] Z. Song, G. Wang, Y. Yu, T. Chen, *et al.*, “Digital identity verification and management system of blockchain-based verifiable certificate with the privacy protection of identity and behavior”, *Security and Communication Networks*, vol. 2022, 2022.

[6] D. R. Malik, D. A. Kataria, and D. N. Nandal, “Analysis of digital wallets for sustainability: A comparative analysis between retailers and customers”, *International Journal of Management*, vol. 11, no. 7, 2020.

[7] S. Chuhan and V. Wojnas, “Designing and evaluating a resident-centric digital wallet experience”, in *International Conference on Human-Computer Interaction*, Springer, 2023, pp. 591–609.

[8] B. Otto, “A federated infrastructure for european data spaces”, *Communications of the ACM*, vol. 65, no. 4, pp. 44–45, 2022.

[9] D. Pöhn, M. Grabatin, and W. Hommel, “Modeling the threats to self-sovereign identities”, *Open Identity Summit 2023*, 2023.

[10] M. M. Alam, A. E. Awawdeh, and A. I. B. Muhamad, “Using e-wallet for business process development: Challenges and prospects in Malaysia”, *Business Process Management Journal*, vol. 27, no. 4, pp. 1142–1162, 2021.

[11] M. Chen *et al.*, “Research on identity authentication of IoT devices based on blockchain”, in *2022 IEEE 4th International Conference on Power, Intelligent Computing and Systems (ICPICS)*, IEEE, 2022, pp. 458–465.

[12] M. A. Hassan and Z. Shukur, “Device identity-based user authentication on electronic payment system for secure e-wallet apps”, *Electronics*, vol. 11, no. 1, p. 4, 2022.

[13] S. Gajek, H. Löhr, A.-R. Sadeghi, and M. Winandy, “Truwallet: Trustworthy and migratable wallet-based web authentication”, in *Proceedings of the 2009 ACM workshop on Scalable trusted computing*, 2009, pp. 19–28.

[14] J. Sedlmeir, R. Smethurst, A. Rieger, and G. Fridgen, “Digital identities and verifiable credentials”, *Business & Information Systems Engineering*, vol. 63, no. 5, pp. 603–613, 2021.

[15] R. Dhamija and L. Dusseault, “The seven flaws of identity management: Usability and security challenges”, *IEEE Security & Privacy*, vol. 6, no. 2, pp. 24–29, 2008.

[16] B. Zwattendorfer, T. Zefferer, and K. Stranacher, “An overview of cloud identity management-models.”, *WEBIST (1)*, pp. 82–92, 2014.

[17] B. Pfitzmann and M. Waidner, “Privacy in browser-based attribute exchange”, in *Proceedings of the 2002 ACM workshop on Privacy in the Electronic Society*, 2002, pp. 52–62.

[18] N. Selvanathan, D. Jayakody, and V. Damjanovic-Behrendt, “Federated identity management and interoperability for heterogeneous cloud platform ecosystems”, in *Proceedings of the 14th International conference on Availability, Reliability and Security*, 2019, pp. 1–7.

[19] S.-H. Kim, S.-R. Cho, and S.-H. Jin, “Context-aware service system architecture based on identity interchange layer”, in *2008 10th International Conference on Advanced Communication Technology*, IEEE, vol. 2, 2008, pp. 1482–1486.

[20] A. Abraham, C. Schinnerl, and S. More, “SSI strong authentication using a mobile-phone based identity wallet reaching a high level of assurance.”, in *SECRYPT*, 2021, pp. 137–148.

[21] T. South and R. Mahari, “Justice in a vaccum?”, 2023.

[22] A. Kudra, “Self-sovereign identity (SSI) in deutschland: Projekte mit strahlkraft für die globale community”, *Datenschutz und Datensicherheit-DuD*, vol. 46, no. 1, pp. 22–26, 2022.

[23] M. Gaedke, J. Meinecke, and M. Nussbaumer, “A modeling approach to federated identity and access management”, in *Special interest tracks and posters of the 14th international conference on World Wide Web*, 2005, pp. 1156–1157.

[24] N. Jahnke and B. Otto, “Data catalogs in the enterprise: Applications and integration”, *Datenbank-Spektrum*, pp. 1–8, 2023.

[25] V. Siska, V. Karagiannis, and M. Drobics, “Building a dataspace: Technical overview”, 2023.

[26] S. Santhar, A. Das, M. Thomas, and S. Prasad, “Self-describing digital assets and their applications in an integrated science and engineering ecosystem”, in *Accelerating Science and Engineering Discoveries Through Integrated Research Infrastructure for Experiment, Big Data, Modeling and Simulation: 22nd Smoky Mountains Computational Sciences and Engineering Conference, SMC 2022, Virtual Event, August 23–25, 2022, Revised Selected Papers*, Springer Nature, 2023, p. 274.

[27] K. Bernsmed, M. G. Jaatun, P. H. Meland, and A. Undheim, “Security SLAs for federated cloud services”, in *2011 Sixth International Conference on Availability, Reliability and Security*, IEEE, 2011, pp. 202–209.

[28] M. R. Ahmed, A. M. Islam, S. Shatabda, and S. Islam, “Blockchain-based identity management system and self-sovereign identity ecosystem: A comprehensive survey”, *IEEE Access*, vol. 10, pp. 113 436–113 481, 2022.

[29] P. N. Mahalle and P. N. Railkar, *Identity Management for Internet of Things*. CRC Press, 2022.

[30] M. Forti, “A legal identity for all through artificial intelligence: Benefits and drawbacks in using AI algorithms to accomplish SDG 16.9”, in *The Ethics of Artificial Intelligence for the Sustainable Development Goals*, Springer, 2023, pp. 253–267.

[31] K. Lampropoulos, N. Kyriakoulis, and S. Denazis, “Identity management through a global discovery system based on decentralized identities”, *arXiv preprint arXiv:2212.02185*, 2022.

[32] R. V. Manikandan, K. Gurunathan, D. Ravindran, M. Sanjai, and V. P. Raja, “An novel algorithm for cloud secure storage using cloud dispersion and block chain system”, in *2023 4th International Conference on Signal Processing and Communication (ICSPC)*, IEEE, 2023, pp. 372–376.

[33] S. Banabilah, M. Aloqaily, E. Alsayed, N. Malik, and Y. Jararweh, “Federated learning review: Fundamentals, enabling technologies, and future applications”, *Information Processing & Management*, vol. 59, no. 6, p. 103 061, 2022.

[34] C. Cuijpers and J. Schroers, “eIDAS as guideline for the development of a pan european eID framework in FutureID”, 2014.

External Representations of Semantically Rich Content in Complex Systems Using Loose Coupling

Hans-Werner Sehring
Nordakademie
Elmshorn, Germany
e-mail: sehring@nordakademie.de

***Abstract*—Content-based software systems like websites and online shops are based on multiple components that collaborate in different ways while serving users. In recent years, the typical architecture of solutions centered around CMSs changed from monolithic to loosely coupled systems. Current approaches are called “composable architecture” or “composable commerce” because they focus on substitutability of components that provide a specific service. Data exchange between system components takes place in an external format that conforms to a system-wide agreed schema. Content Management Systems (CMSs) are one central component of a content-based system. CMSs manage and publish meaningful content. Such content is represented by data, but it is not processed under fixed semantics. However, collaborating systems require a consistent interpretation of data as content on both ends in order to preserve meaning. We argue that such a consistent interpretation requires mappings between the content models underlying CMSs and the data models that are used for communication, and that these mappings, therefore, must be shared by all components of the system. In order to justify this claim, we compare the expressiveness of plain data formats with that of content modeling languages, and we study mappings between them. In this paper, we use JSON and JSON Schema as typical examples of external data representations. We discuss content models using the example of the Minimalistic Meta Modeling Language (M³L). Our initial research shows that schemas for data exchange should be tightly linked to content models in order to not only represent content as data, but also to allow for consistent interpretations of content.**

Keywords*—*content model; data schema; schema mapping

I. INTRODUCTION

Content Management Systems (*CMSs*) are an established tool for (in particular online) content publication. They are software systems that incorporate various functions for content creation, editing, management, (automated) document creation based on layouts, and document delivery. Over time, many CMS products started integrating additional functionality to keep up with emerging requirements. At the same time, such products became increasingly complex because many of them incorporate new functions in a monolithic way.

Since they often provide a comprehensive software infrastructure comparable to an application server, many content management solutions are built using a CMS as a platform. Custom code is integrated into the CMS, making the overall solution an even larger monolith. This approach is often suitable for purely content-based functionality.

In recent years, an opposite trend has taken hold under the name *headless CMS*. Such CMSs basically focus on basic content creation, editing and management functions. Content is published via a delivery service that makes “pure” content accessible in the form of *Application Programming Interfaces* (*APIs*). All additional services are provided by separate software components. This includes document preparation and delivery that is implemented outside of a headless CMS. Components are combined using a *composable architecture*, also called *composable commerce*.

Though the idea of using simple interfaces based on a data exchange format is appealing, it constitutes an “impedance mismatch” with rich content structures as employed by capable CMSs. Ideally, a CMS provides various means of structuring content. Many allow defining a *schema* or *content model*. Such a schema is, on the one hand, used to provide type safety to functions handling content, and on the other it constitutes the basis to capture the meaning of content. To make use of structure and meaning assigned to content, content structure and semantics defined by content models need to be preserved in external representations, and they are used as a basis to map content to an external form.

In this article, we argue that expressive content models are required to globally describe the meaning of content so that content is correctly represented as data and data is interpreted as content. The discussion in this article extends the presentation in [1].

With the landscape of digital communication solutions becoming more complex, there is an increasing number of services that integrate data and services for other entities than structured content – media files, customer data, product data, etc. The services need to interface with CMS solutions. Systems typically require application-specific integrations (see, for example, [2]). These integrations make systems that rely on data exchange with a centralized monolithic platform overly complex since they have to deal with a variety of data exchange formats and different entity lifecycles.

We discuss complex content-based systems and the various perspectives on content in such systems in Section II.

Systems that incorporate CMSs using APIs typically are built following microservice architectures. These consist of multiple services that provide one functionality each, with the CMS providing content as one of those services. System properties are established by service orchestration in the overall architecture.

APIs for access to content consist of service signatures and of structured content representations that are used as input and output parameters. Content representations typically focus on

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

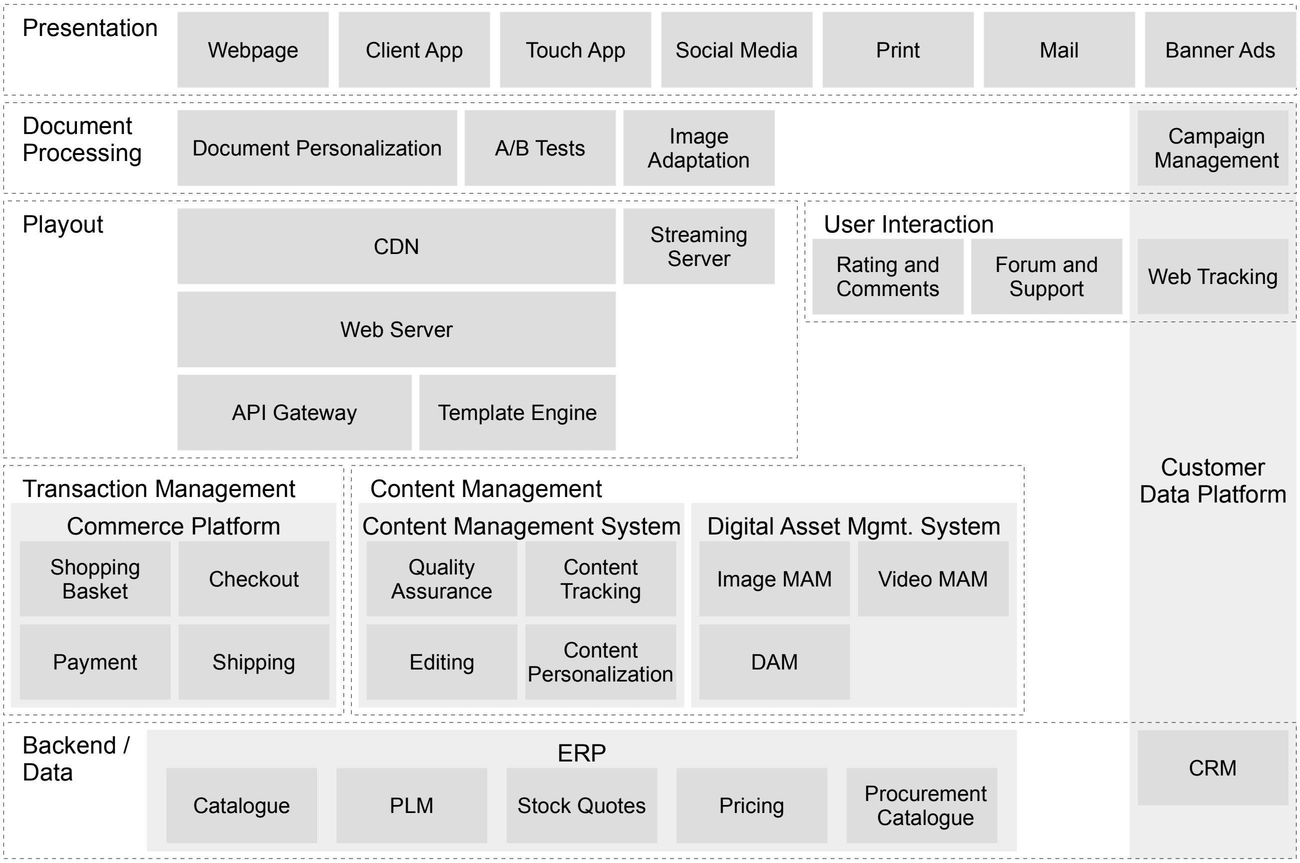


Figure 1. High-level Architecture [3].

structured content – mainly textual content and descriptions of unstructured content. Unstructured content, be it provided by a CMS or a Digital Asset Management system, is typically transferred in some binary format.

RESTful APIs are a current de-facto standard for communication between distributed CMS components. The *JavaScript Object Notation* (*JSON*) is the usual language chosen to represent (structured) content. Section III names typical aspects of APIs defined this way.

In Section IV, we introduce the Minimalistic Meta Modeling Language (M³L) as an example of a rather powerful content modeling language.

We use the M³L's capabilities for binding to external representations to study some aspects of interfaces for content access and interchange. In particular, we demonstrate different cases of JSON generation and parsing in Section V and discuss general differences of custom generated JSON and such generated by content models formulated in the M³L in Section VI.

We conclude the paper in Section VII.

II. Interfaces in Content-Based Systems

Content in the content management sense is found in a class of applications that enables digital communication. In a commercial setting, it primarily addresses the communication between companies and their customers. These interact using marketing websites, online shops, customer support systems, mobile apps, and similar mass communication facilities.

Content used to be of central interest because it provides the language of companies used in mass communication. This includes pragmatics, tonality, etc. With a tighter integration of communication means on digital communication platforms, other entities received equal interest, first and foremost customers (or, more precisely: the relationship of customers to companies).

With growing digital communication platforms, the need to exchange content between system components becomes ever more immanent.

A. Content-Based Information Systems Architecture

Different communications channels that are centered around customers and content are often integrated in growing content-based information systems. The increasing number of system components that contribute to such platforms and the additional need of integrating content and content-centric processes pose a challenge that companies have to master today.

Single communication requirements are met by specific systems, and new approaches and systems continuously emerge. To guide the creation of information systems from such base systems, reference architectures have been formulated.

Figure 1 outlines such an architecture for a website infrastructure that is typical for online shops, for example.

The architecture introduces layers in which customer-facing systems are placed on the top of the figure and internal, more technical components are located at the bottom.

The dotted lines in the figure are used to show functional clusters of components that in concert provide some service.

The *presentation* area on the top of the figure lists some (digital) communication channels. A company's website is often at the core of a communication strategy. Other channels may co-exist, but at best the channels work together in an omni-channel strategy. For example, a newsletter sent via mail may hint users to new content on the website, or banner ads redirect users to the website.

The basic *content management* functionality is located at the center of Figure 1. Besides CMSs there are digital asset management (DAM) and multimedia asset management (MAM) systems located in this functional area.

Content is delivered on the communication channels by creating and distributing documents with that content during *playout*. Typically, documents are generated from templates into which content is filled in. Webservers distribute the documents via a caching layer.

Even though the preparation of content for document creation is a content management task, documents are post-processed in some cases. *Document processing* functions often are Cloud services that leverage the core systems from optimization tasks.

Besides content management, a *transaction management* area drives processes like sales and logistics. These are different in nature from content-based processes, but the two closely interact.

Other means of *user interaction* allow users to provide own content, not only to consume the existing one. User-generated content calls for specific handling that differs from content management. User interaction also is a very valuable source of information on the users.

At the core of a content-based system, some data-driven *backend* services provide the backbone for all business processes.

Inside the dotted areas, there are components that provide single functions each. In component architectures, the components are typically realized by system products. For example, the area of the content management system is implemented by one CMS. Often, it will also incorporate functionality of the *playout* and the *document processing* area. In (micro) services architectures, it is more likely that single functions will be realized by distinct services. Then the larger functional clusters are implemented by orchestrated services.

B. Content Semantics

Content is a term for which there is no uniform definition. In this article, we will not provide a formal definition either. But we assume a certain notion of the term content as a basis, and we distinguish content from data.

For the purpose of the discussion in this article, data is meant to be any formal, digital representation used for storage, transmission, and processing.

Content may be represented by data, but it is something valuable and purposeful, where the purpose is some aspect of digital communication between system users.

Content is also distinguished from documents which are presentations of content. Documents are also determined by a layout and other visual properties that define how content is represented. Depending on the application, the dividing line may vary. For example, the layout of the text of a news article may be a presentation issue, where the position of text in a figure carries semantics.

With these notions, content conveys something people want to communicate about. It is about domain entities that are the subject of some communication. The information on and views of the domain entities may vary and may be subjective. Still, content is meant to capture domain semantics attached to it.

Content follows the semantics of the application domain, data a formal semantics. Both are required in communication. For example, products and services are of central importance on commercial websites. They are described by both content that represents people's views and by data that provides objective facts and figures. While content needs to be interpreted by people with similar views, data can formally be interpreted based on the semantics of some formal domain like mathematics. Product descriptions on commercial websites exhibit this twofold description feature. Marketing texts, legal texts, etc. are provided as content. Product data, like price, physical properties, shipping information, etc. is described by data.

Similar to a data model that conforms to a data schema, a content model is an abstraction of a content set. A content model serves two purposes. Like a data schema, it describes how content is formed. But it also captures some domain semantics, for example, by describing how certain domain entities are meant to be represented in content.

C. Content Management Interfaces

Websites and mobile apps are an important user interface to information sources, and they constitute a means for companies to get in touch with their customers. These applications are based on content that is presented to users in a suitable form.

Contemporary implementations of CMSs, online shops, campaign management solutions, and other content-based systems deal with content in various places: databases, application code, user interfaces, remote calls, URL formatting, HTTP request handling with content lookup and caching, tracking, targeting, campaign attribution, and many more. Figure 2 gives a rough overview.

Technically, content is stored, processed, and transferred as data. The multitude of content applications is reflected in diverse interfaces between the different components that together form a content-based system. Different kinds of interfaces are in place.

Editors collaborate on centrally stored content. To support this collaboration, instances of the content editor tool use a synchronous interface to the central CMS. It is characterized

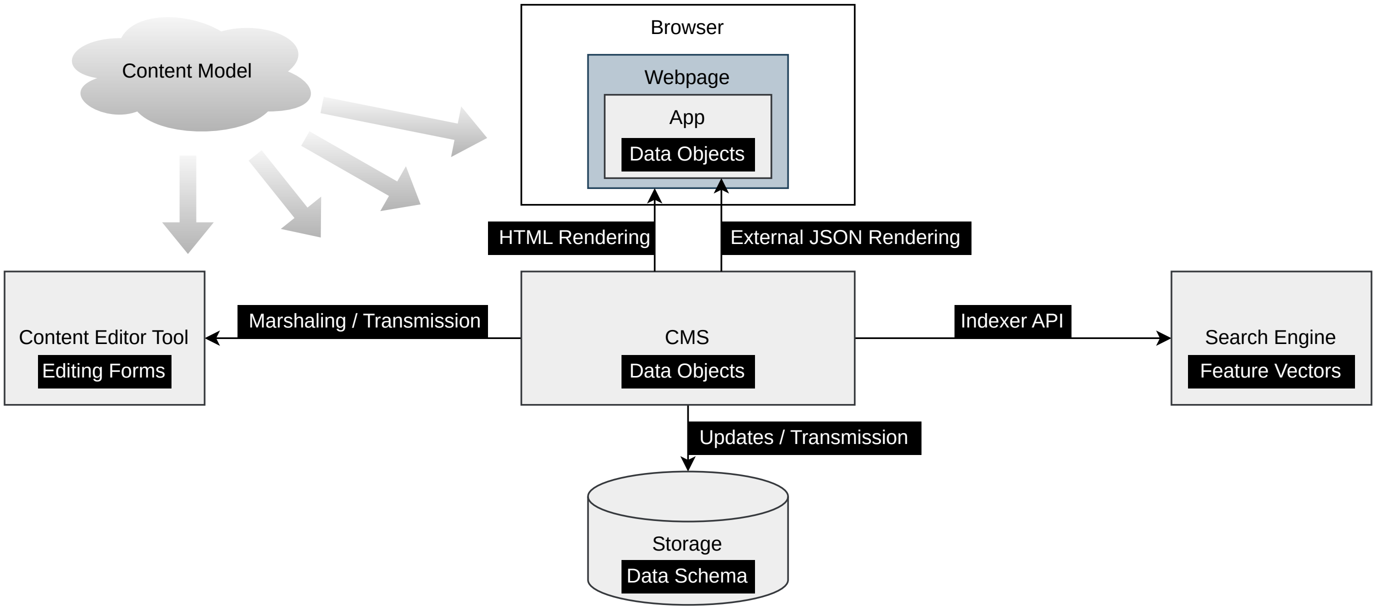


Figure 2. Content model references in a typical content management system.

by loose coupling with message passing, so that editors are notified about changes, and concurrent work is enabled by locking strategies.

Tightly coupled systems like the search engine, on the other hand, exchange copies of content. A search index is built up from content that is analyzed according to its structure.

For publication, documents are created from content. For example, web pages are created to publish content on the web. In particular in this scenario, document generation is called content rendering. It differs depending on the kind of generation used. Server-side rendering, where documents are created by the service provider, works in a synchronous fashion, because documents have to be completely generated before they can be shipped. Client-side rendering, on the other hand, may improve user experience by dynamically loading and rendering content in an asynchronous fashion. Hybrid approaches are usually in place today, and these need both kinds of interfaces to the central CMS.

Each of the indicated functionalities is related to the underlying content model, and all share a common notion of both this model and all content constellations. Or, viewed the other way round, a content model defines multiple interfaces that are consumed by different audiences, for example in a CMS:

- editors that are guided by the editing tool when entering content into forms, and that are supported during quality assurance of webpages,
- an editor-in-chief sees content before publication during quality assurance,
- compliance officers also receive pre-final presentations plus differences from previous versions to conduct legal and compliance reviews,
- application programmers that customize the CMS (services, editor, search engine),
- application programmers that develop client-side apps (JavaScript apps, mobile apps),
- template programmers that implement the rendering of content into documents,
- search engine optimization (SEO) managers who look after content descriptions, and
- last but not least users who think in content categories (product, contact, etc.) while browsing and reading, and who may use a refined search function.

In fact, each of the roles uses more than one interface, and all need to agree on the conceptual content model in order to use and serve the interfaces correctly. In particular, content needs to be encoded as data, and data need to be interpreted as content. Figure 2 names some of the technical interfaces where this applies.

A common content model is not only required for technical reasons of using APIs. Also, there needs to be a globally agreed domain semantics of content as argued in the previous section. For example, an editor maintains the content that is included in documents. For the user of the system to perceive what the editor had in mind, the editor's interface, the schema for content storage, and the APIs used by template programmers all have to be in line with a common understanding of content semantics.

Therefore, for coherence in content-based applications, there is a need for central models that are consistently implemented, or schemas and code for such systems are generated from such models. The upper left of Figure 2 symbolizes this.

III. STANDARDS: RESTFUL APIS, JSON, AND GRAPHQL

Approaches for (remote) APIs and their implementations are of general interest since the advent of distributed systems. After a series of technological approaches, a current de-facto standard for online interfaces of CMSs has emerged from REST, JSON, and GraphQL.

A. *RESTful APIs*

Representational State Transfer (REST) was proposed by Fielding as the principle of communication in the Internet [4]. It calls for stateless servers and clients that handle state between requests. In conjunction with URLs that represent services calls, the definition of so-called RESTful APIs allows defining simple APIs for Web-based services.

Such APIs consist of service call signatures composed of an HTTP method and a URL that specify the service to be used and the input parameters. The response to a service call is a regular HTTP response. A typical response format for structured data is JSON as discussed in the subsequent section.

RESTful APIs can be implemented with existing Web technologies, for example, typical software libraries available for all relevant programming languages and existing software components to build a service infrastructure.

B. *Content Interchange with JSON*

JSON is an object language for JavaScript, allowing to formulate JavaScript object instances, where *instance* refers to data contained in object properties, not any internal object state. It can be used for data storage, transmission, and aggregation.

JSON is typically used as a response format of RESTful interfaces. It provides a simple means of structuring data with typical collection types, and encoding of data as character strings.

Most API-based CMSs use JSON to distribute content. They typically do so by representing content in a straight-forward manner using the structuring means and primitive data types of JSON.

Internally, CMSs allow the definition of content models that describe content. Such models are the basis for describing content, for content editing, and also for JSON generation.

Depending on the kind of CMS, such content models are more expressive than data models in JSON. In structured CMSs, content models are used to capture domain semantics attached to content. In contrast, *Digital Experience Platforms* (*DXPs*) focus on associating content with visual layouts. Content models describe visual building blocks in this case.

Document rendering presents content in visible form for consumption, for example, in the form of HTML files. In API-based CMS solutions, rendering is performed by external rendering engines (on client-side or on server-side). The rendering process is driven by *templates* that define how to layout content. Template code makes use of knowledge about the meaning of content to be represented, either the domain semantics (CMS) or the kind of visual building block (DXP).

The external form of content in a JSON representation is rather generic, though. JSON can be generated in an application-specific form, but basically contains structured data. The representation of content in JSON and interpretation from that format rely on consistent code on both producer's and consumer's side. Such interpretation cannot rely on JSON representations of content alone.

C. *JSON Schema Languages*

JSON as a format is appealing because of its simplicity combined with reasonable expressiveness. It was defined merely for the description of single records of data. Many applications call for a schema, though, that describes how classes of data are structured.

Several schema languages have been defined for JSON, most prominently *JSON Schema* [5]. Another proposal for a JSON schema language is JSound [6]. Other approaches are Joi [7] for JavaScript applications and Mongoose [8] for configurations of the database system MongoDB.

In this paper, we use JSON Schema for the discussion of schema properties.

D. *GraphQL*

GraphQL is a query language for structured data. Queries select data from a database, and they describe a JSON format in which the response to a query is expected.

JSON-based APIs often face the problems of "underfetching" (there is no call that delivers all data required in a situation so that a sequence of related remote calls is required) and "overfetching" (a service delivers too much data in a situation, so that too much data is transmitted and a client has the task of selecting the required data).

Using GraphQL as a service interface language helps avoiding these problems. Clients can select the exact dataset they need in a specific situation. On the other hand, GraphQL is dynamic by nature leading to more computation and less options for caching of results.

IV. A SHORT INTRODUCTION INTO THE MINIMALISTIC META MODELING LANGUAGE (M³L)

For the discussion in this paper, we use the M³L since it proved to be a suitable language for the modeling of various aspects of content management. To this end, we briefly introduce the M³L, and we present some exemplary base models for content management and for content interchange based on RESTful APIs.

A. *A Short Introduction into the M³L*

In this section, we briefly introduce the M³L by highlighting those features that are central to the underlying experiments.

The basic M³L statements are:

- **`A`**: the declaration of or reference to a *concept* named *A*
- **`A`** `is a` **`B`**: refinement of a concept *B* to a concept *A*. *A* is a *specialization* of *B*, *B* is a *generalization* of *A*.
- **`A`** `is a` **`B`** `{` **`C`** `}`: containment of concepts. *C* belongs to the *content* of *A*, *A* is the *context* of *C*.
- **`A`** `|=` **`D`**: the *semantic rule* of a concept. Whenever *A* is referenced, actually *D* is bound. If *D* does no exist, it is created in the same context as *A*.
- **`A`** `|-` **`E F G.`**: the *syntactic rule* of a concept that defines how a string is produced from a concept, respectively how a concept is recognized from a string.

When the representation of *A* is requested, it is produced by a concatenation of the strings produced out of *E*, *F*, and *G*. When no syntactic rule is defined, a concept is represented by its name. Vice versa, an input that constitutes the name of a concept without a syntactic rule leads to that concept being recognized.

If a concept that is referenced by one of the statements exists or if an equivalent concepts exists, then this one is bound. Otherwise, the concept is created as defined by the statement.

Existing concepts can be redefined. For example, with the definitions above, a statement

```
A is an H { C is the I }
```

redefines *A* to have another generalization *H* and *C* (in the context of *A*) to have *I* as its only generalization.

Every context constitutes a *scope*. A redefinition of a concept in a context is only applied in that context. When a redefinition of a concept takes place in another context as the original definition, we call that redefinition a *derivation*.

The concepts that are defined by such statements are evaluated when used. Context, specializations, and semantic rules are employed for *concept evaluation*. A concept evaluates to the result of its syntactic rule, if defined, or to its *narrowing*. A concept *B* is a narrowing of a concept *A* if

- *A* evaluates to *B* through specializations or semantic rules, and
- the whole content of *A* narrows down to content of *B*.

To evaluate a concept, syntactic rules and narrowing are applied repeatedly.

Given the sample M^{3}L statements:

```
Person { Name is a String }
PersonMary is a Person { Mary is the Name}
PersonPeter is a Person {Peter is the Name
                          42    is the Age
}
```

the result of an additional statement

```
Person { Peter is the Name 42 is the Age}
```

is *PersonPeter* since *PersonPeter* is specialization of *Person* and its whole content matches. The statement

```
Person { Mary is the Name 42 is the Age }
```

is not evaluated further. It does not match *PersonPeter* since *Name* has a different specialization, and it does not match *PersonMary* since that has no matching content concept called *Age* or *42*.

B. Basic Content Management and Document Rendering

The M^{3}L is universal and has many applications. Among other modeling tasks, it has proven useful to describe content as lined out in, for example, [9]. This applies both to content models as well as content items since the M^{3}L does not distinguish model layers, such as type and instance.

For example, with a content model like:

```
Article is a Content {
  Title is a String
  Text  is a FormattedString }
```

according content can be created:

```
NewsArticle123 is an Article {
  "Breaking News"           is the Title
  "This is a report on ..." is the Text }
```

For textual formats, like HTML and JSON, documents can be rendered from content through syntactic rules of content as introduced in the previous subsection. On the level of the content model, syntactic rules describe document templates, on the content item level they render single document instances.

For the sample content definitions above, a JSON template for a piece of content of type *Article* may look like:

```
Article |- "{\"title\":\"" Title
           "\",\"text\":\"" Text "\"}"
```

This syntactic rule produces JSON output for the concept *NewsArticle123* from above:

```
{"title":"Breaking News","text":"This is a
report on ..."}
```

The syntactic rule defines a JSON structure into which the concepts from the content are integrated. These may themselves evaluate to content strings of embedded JSON structures.

Please note that, for example, *{\"title\":\"* is a valid concept name, as is *\"}*. Since new concepts are declared the first time they are referenced, and because they syntactically evaluate to their name by default, they can be used like string literals. The concept name *\"* is an escape sequence for the quote character (not a quote sign for identifiers).

V. PRODUCING JSON USING THE M^{3}L

As outlined in the preceding section, the M^{3}L can serve as an example of a an expressive content modeling language. For API-driven content distribution, structured content needs to be represented in an external form. In state-of-the-art services, this external form is JSON.

The same holds for JSON generation. JSON Schema allows defining valid forms of JSON structures so that content can be transferred in a reliable manner. It is not expressive enough by itself, however, to recover equivalent content on the receiver's side. Custom code is required to generate JSON out of rich content structures. Appropriate code that shares the same conception of content is required to interpret JSON data.

Schema design for JSON requires careful consideration. Even finding sample instances for a given schema is a non-trivial task since semantics is scattered over a set of definitions and constraints [10].

JSON Schema provides various ways of defining and relating schemas. There are multiple ways of expressing equivalent schemas and equivalence cannot generally be proven [11].

One way of sharing content concepts between sender and receiver is to have a common content model and mappings to and from external representations. We exemplify this by utilizing the capabilities of the M^{3}L for some sample constructs.

A. Defining Lexical Rules for JSON

M³L's lexical rules can produce JSON code out of concepts as sketched in Section IV-B.

The M³L does not distinguish between types and instances of content, let alone other incarnations (materializations, metatypes, etc.). This distinction is, however, required in classical approaches as JSON and JSON Schema.

In addition to the above sample rules that generate JSON, the lexical rules of other concepts may produce JSON Schema. See the following simple rules for the content example. A M³L definition:

```
Article |-
"{\"type\":\"object\",\"properties\":{\""
   Title "\":{\"type\":\"string\"},"
   Text "\":{\"type\":\"string\"}}}"
```

results in the production of the following JSON Schema definition:

```
{ "type": "object",
  "properties": {
    "Title": { "type": "string" },
    "Text":  { "type": "string" }}}
```

Lexical rules for both JSON and JSON Schema require to distinguish between schema and instances. Contextual definitions allow defining both layers for a concept. The decision between schema and instance has to be made explicitly, for example, by providing separate sets of syntactic rules in different contexts:

```
SchemaRules { Article |- ... }
InstanceRules { Article |- ... }
```

The distinction between schema and instance level is atypical for M³L applications. Usually, concepts my play both roles. Therefore, it is possible that the same concept will sometimes be represented by JSON (as an instance) and sometimes as JSON Schema (when it contributes to the structure of another concept)

In any case, a fair amount of extra code is required to state the obvious lexical rules per concept. It is approximately the same effort like providing custom mappings in software.

The effort of mapping an internal content model to its external forms is beneficial, though, to be able to recover the semantics of content. This way, schema definitions contribute to the exchange of meaningful content. In the subsequent subsections, we compare the modeling capabilities of JSON Schema and the M³L for the generation of JSON representations of content.

B. Basic Model Mapping from M³L to JSON

Simple M³L expressions that represent content instances can be expressed in a straight-forward manner as outlined by the content example. Some information is lost in the JSON representation, though. In the example above, the concept name *Article* is not communicated.

Such concept information may be reflected in dedicated properties. But more information on the content is lost if we add content types and descriptions, for example in M³L:

```
Person {
  FirstName is a String
  LastName  is a String
  Address }
Address {
  Street is a String
  City   is a String }
JohnSmith is a Person {
  John               is the FirstName
  Smith              is the LastName
  JohnSmithsAddress is the Address {
    "Main Street" is the Street
    Lincolnshire  is the City } }
```

Syntactic rules may produce the following JSON:

```
{ "FirstName": "John",
  "LastName": "Smith",
  "Address": { "Street": "Main Street",
               "City": "Lincolnshire" } }
```

The intended data structure can be defined by means of JSON schema that is also generated from the content concepts, for example, as follows:

```
{"title": "Person",
 "type": "object",
 "properties": {
  "FirstName": {"type": "string"},
  "LastName": {"type": "string"},
  "Address": {"$ref":"#/$defs/Address"} },
 "$defs": {
  "Address": {
   "type": "object",
   "properties": {
    "Street": {"type": "string"},
    "City": {"type": "string"} } } } }
```

Here, the concept names *Person* and *JohnSmith* are not present in JSON. The content name *JohnSmithsAddress* is also missing; the "type" name *Address* is used instead.

Note that information is distributed over two structures, instance and schema, and declared in different languages. A JSON (instance) file does not make reference to the schema it is intended to comply with. Therefore, the matching schema has to be found by distinct means. Names – concept names in the case of the M³L – are not included in JSON, but are required for schema selection (*Person* in the above example).

Additional information that relates data to its schema has to be added on top of the standard data formats. One option is the use of a kind of envelope structure which adds type information to data, for example,

```
{ "JohnSmith": { "Firstname": "John",... },
  "type": "Person" }
```

For a second option we may introduce well-defined property names under which we give metadata as part of the data in a JSON structure. For example, assume that the property names *$name* and *$type* are defined to hold such metadata:

```
{ "$name": "JohnSmith",
  "$type": "Person",
  "Firstname": "John", ... }
```

In order to generate two external forms – JSON and JSON Schema – out of one integrated internal content representation, two lexical rules are required as mentioned in Section V-A. When parsing JSON on the receiver's side, the unrelated files need to be recombined in a content representation. JSON (Schema) provides no means to do so.

C. Capturing Type Variations

Variants of content are commonly found in CMSs since one schema alone typically does not cover all aspects under which content is used for communication. Few CMSs cover variations explicitly in content models. The M^{3}L, however, allows reflecting variants by means of concept refinement and by contextualization.

Consider concepts modeled after an example from [12]:

```
Address {
  street_address" is a String
  city            is a String
  state           is a String
  Type }
BusinessAddress is an Address {
  Business is the Type
  Department is a String }
ResidentialAddress is an Address {
 Residential is the Type }
```

An example from [12] reflects the above M^{3}L definitions:

```
{ "type": "object",
  "properties": {
    "street_address": {"type": "string"},
    "city": { "type": "string" },
    "state": { "type": "string" },
    "type": {
      "enum": ["residential", "business"]
    }
  },
  "required": ["street_address",
               "city", "state", "type"],
  "if": {
    "type": "object",
    "properties": {
      "type": { "const": "business" }
    },
    "required": ["type"]
  },
  "then": {
    "properties": {
      "department": { "type": "string" }
    }
  },
  "unevaluatedProperties": false }
```

In this example, an address has a type with possible values from the enumeration: *residential* and *business*.

The additional *department* property from the example above can be introduced conditionally using the *"if"..."then"..."else"* construct in JSON schema. This allows representing subtypes. The information that a *BusinessAddress* is an *Address* is lost, however, both on instance and schema level. Therefore, this is not a suitable representation of refinement that conveys semantics.

In the M^{3}L, we have extra concepts for a *BusinessAddress* and a *ResidentialAddress*. Therefore, the *Type* is not actually needed as content. It could be generated into JSON from the concept information.

In fact, M^{3}L would (also) work the other way round: if an *Address* with an extra *Department* is given, it is derived to be a *BusinessAddress*. The derived base type matching of the M^{3}L – and a similar, but typically implicit behavior of typical CMS applications – makes matching JSON data to JSON Schema definitions yet more difficult.

The M^{3}L can make those type variations explicit, though. In the above example, we can assign a *Type* property as required by JSON schema based on the presence of *Department* information:

```
Address {
  street_address is a String
  city           is a String
  state          is a String
} |= Address {
  street_address is the street_address
  city           is the city
  state          is the state
  Residential    is the Type }
BusinessAddress is an Address {
  Department is a String
} |= BusinessAddress {
  street_address is the street_address
  city           is the city
  state          is the state
  Business       is the Type }
ErrorneousAddress is an Address {
  Department  is a String
  Residential is the Type
} |= Error
```

This is not considered a typical M^{3}L application, though.

VI. Comparison of Plain JSON and M^{3}L Constructs

In contrast to typical data schemas, content models are not only concerned with constraints on values, references, and structure, but additionally try to capture some semantics. Furthermore, while data aims at representing one consistent state of entities, content deals with varying forms and utilizations used in communication: different communication scenarios, contexts of users who perceive content, language and other localizations, etc.

This section points out some of the differences in expressiveness of data schemas and content models using the examples of JSON schema and the M^{3}L.

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

A. Subtypes

Type hierarchies allow intensional descriptions of schema elements and are, therefore, found in content models. They are not ubiquitous in data models, though. JSON schema does not feature subtyping.

JSON Schema does have means to express schema variants (*"if"*, *"dependentRequired"*) and to relate different schemas (*"dependentSchemas"*, *"allOf"*, *"anyOf"*, *"oneOf"*).

These can be used to model specializations of data as variants. An example is presented in Section V-C above.

Any forms of refinements (“subtypes”) in JSON weakens the constraints of a JSON Schema since not all properties can be *"required"* or *"additionalProperties"* and *"unevaluatedProperties"* must be allowed - very much as in the M^{3}L.

The M^{3}L as a notion of refinement that does not explicitly distinguish between subtyping and instantiation. In the following definitions, for example,

```
Employee is a Person {
 Salary is a Number }
JohnSmith is an Employee {
 John is the FirstName
 Smith is the LastName
 5000 is the Salary }
```

present a concept hierarchy where *Person* and *Employee* would presumably be handled on the schema level in external representations and *JohnSmith* at instance level. Then *Person* and *Employee* are in a subtype relationship with inheritance.

In JSON Schema, this would be expressed as:

```
{ "title": "Person",
  "type": "object",
  "properties": {
    "FirstName": { "type": "string" },
    "LastName": { "type": "string" },
    "Salary": { "type": "number" },
    "is_employee": { "type": "null" }
  },
  "required": ["FirstName", "LastName"],
  "dependentRequired": {
    "is_employee": ["Salary"]
  } }
```

Again, the *Employee* as a concept is not represented.

B. Single and Multi-valued Relationships

It is quite common in content models to be vague about arity. For example, some pieces of content may typically have a 1:1-relationship, making it unary in the content model. But there are exceptions of n-ary cases that also need to be covered. The M^{3}L allows to define concepts with *is a* and *is the* to take this into account.

A typical data model would define an n-ary relationship, even though in most cases the data are 1:1.

JSON itself allows to easily vary between unary and n-ary properties by simply stating either *"a":"b"* or *"a":["b","c"]*. JSON Schema, though, needs to define arity or to define variations with *"if"..."then"..."else"*.

Consider as an example a person with two addresses:

```
Person {
  FirstName is a String
  LastName  is a String
  Address }
Address {
  Street is a String
  City is a String }
JohnSmith is an Employee {
  John                is the FirstName
  Smith               is the LastName
  JohnSmithsAddress is an Address {
    "Main Street" is the Street
    Lincolnshire  is the City }
  JohnSmithsOffice  is an Address {
    "High Street" is the Street
    Lincolnshire  is the City } }
```

A JSON structure reflecting this content is:

```
{ "FirstName": "John",
  "LastName": "Smith",
  "Address": [
    { "Street": "Main Street",
      "City": "Lincolnshire" },
    { "Street": "High Street",
      "City": "Lincolnshire" }
  ] }
```

Though this is a small change to the JSON structure, it has to be explicitly foreseen in JSON Schema. It is not as easy to vary between one or multiple addresses (in this example) as it is in content models like the M^{3}L or the Java Content Repository [13].

C. Content Conversions and Computed Values

It is common for content models to not only contain content itself but also descriptive information about the content (sometimes referred to as metadata).

For example, a simple data property like

```
{"price": 42}
```

requires additional information to be interpreted correctly (the currency, for example). In simple data models, there is an additional documentation that establishes an agreement on how applications should deal with the data. The possibility to state the unit of measurement is typically found in *Product Information Management systems*.

In these cases, the information needs to be stated explicitly, as it is done in typical master data management systems:

```
{"price": {"value":42, "currency":"€"}}
```

Such a record allows a mutual understanding of the value. It prevents an easy mapping from JSON to a numeric *price* variable, though.

As a slight improvement, values should be replaced by named concepts. The M^{3}L captures meaning by defining relevant concepts. For example, a concept like *EuroCurrency* as a refinement of a concept *Currency* would be used instead of the string value *€*.

On top of descriptive information on content, a content model may also define a limited set of computational rules in order to define consistent arithmetics.

The M³L is expressive enough to define some (symbolic) computation. Assume, for example, a concept *Integer*, concrete "instance" concepts like *100*, and concepts describing computations like *FloatDivision*, the division of numeric values.

On the basis of such definitions, it is possible to state conversion rules like the following:

```
Price {
  Value is a FloatNumber
  Currency }
PriceInEuro is a Price {
  € is the Currency }
PriceInEuroCents is a Price {
  Value is an Integer
  Cents is the Currency }
|= PriceInEuro {
  Value is a FloatDivision {
    Value is the Dividend
    100   is the Divisor } }
```

These sample definitions describe (on schema level) how values are converted so that all clients using this model share the same arithmetics.

D. General Variations

The M³L provides extra flexibility by both considering (internal) definitions and (external) representations. Other approaches like CMSs handle these two aspects separately.

By overriding lexical rules in the M³L, the marshaling format can deviate from the schema. These rules are restricted to very simple grammar rules, though, in order to work both as producers as well as recognizers.

For example, a definition:

```
Person {
  firstName is a String
  lastName  is a String }
|- "{\"name\":\"" firstName " " lastName
   "\"}" .
```

combines first name and last name into one name field. When reading data, it will just break up the name value at the first whitespace.

The simple rules suffice in some situations. In very simple cases, they allow, for example, to adapt legacy JSON or to provide backwards compatibility to previous schema versions. They are not capable of full parsing of input data.

VII. CONCLUSION

We conclude with a summary and an outlook.

A. Summary

We compare rich content models – using the example of the modeling capabilities of the M³L – with typical data schemas, in particular JSON Schema. We conclude that models for meaningful content cannot adequately be expressed by data schemas alone.

JSON became a de-facto standard for content exchange. We present examples showing that the currently evolving schema language, JSON Schema, is not sufficient for content modeling in its current form.

B. Outlook

Additional research is required to identify the full expressiveness required to define external representations of content for modern content management approaches. This will guide future investigations towards a suitable set of modeling capabilities for marshaling formats.

The M³L is not intended to be a data schema language. Therefore, it lacks some features of such languages. It will be an experiment, though, to define a M³L derivative that is able to serve as an alternative schema language for JSON.

ACKNOWLEDGMENT

The author thanks numerous colleagues, partners, and clients for fruitful discussions on various topics centered around digital communication. The Nordakademie is acknowledged for the chance to continue work in the field.

REFERENCES

[1] H.-W. Sehring, "On the Generation of External Representations of Semantically Rich Content for API-Driven Document Delivery in the Headless Approach," Proceedings of the Fifteenth International Conference on Creative Content Technologies, CONTENT 2023, ThinkMind, 2023, pp. 17–22.

[2] H.-W. Sehring, "On the integration of lifecycles and processes for the management of structured and unstructured content: a practical perspective on content management systems integration architecture," International Journal On Advances in Intelligent Systems, volume 9, numbers 3 and 4, pp. 363–376, 2016.

[3] H.-W. Sehring, "Architectural Considerations for the System Landscape of the Digital Transformation," Proceedings of the Twelfth International Conference on Creative Content Technologies, CONTENT 2020, 2020, pp. 5–10.

[4] R. T. Fielding, "Architectural Styles and the Design of Network-based Software Architectures," Doctoral dissertation, University of California, 2000.

[5] A. Wright, H. Andrews, B. Hutton, and G. Dennis, "JSON Schema: A Media Type for Describing JSON Documents," Internet Engineering Task Force, 2022.

[6] C. Andrei, D. Florescu, G. Fourny, J. Robie, and P. Velikhov, *JSound 2.0*. [Online] Available from: http://www.jsound-spec.org/publish/en-US/JSound/2.0/html-single/JSound/index.html. 2024.3.22

[7] *The most powerful schema description language and data validator for JavaScript*. [Online]. Available from: https://joi.dev/ [retrieved: May, 2023]

[8] *Mongoose*. [Online] Available from: https://mongoosejs.com/. 2024.3.22

[9] H.-W. Sehring, "On Integrated Models for Coherent Content Management and Document Dissemination," Proceedings of the Thirteenth International Conference on Creative Content Technologies, CONTENT 2021, ThinkMind, 2021, pp. 6–11.

[10] L. Attouche et al., "A Tool for JSON Schema Witness Generation," Proceedings of the 24th International Conference on Extending Database Technology. OpenProceedings.org, March 2021, pp. 694–697.

[11] F. Pezoa, J. L. Reutter, F. Suarez, M. Ugarte, and D. Vrgoč, "Finding Data Compatibility Bugs with JSON Subschema Checking," Proceedings of the 30th ACM SIGSOFT International Symposium on Software Testing and Analysis, ISSTA 2021. Association for Computing Machinery, July 2021, pp. 620—632.

[12] M. Droettboom, "Understanding JSON schema," Space Telescope Science Institute, 2023.

[13] D. Nuescheler et al., "Content Repository API for Java™ Technology Specification," Java Specification Request 170, version 1.0, May 2005.

Reliable Color Recognition in Images by Using a Modified HSV Algorithm

Samuel Kosolapov
Department of Electronics
Braude College of Engineering
Karmiel, Israel
e-mail: ksamuel@braude.ac.il

Abstract— **There are several situations in which the HSV (Hue, Saturation, Value) algorithm is a natural choice for executing color recognition in a specific image. Unfortunately, whereas Value and Saturation can be calculated in any situation, Hue cannot be calculated when the specific pixel is gray. Practically, for dark regions, nearly gray regions, and for overbleached areas, the calculation of the Hue must be considered as non-reliable. Inherent to digital camera noise, makes the calculations of Hue in the above situations problematic. In an attempt to provide more control during color manipulations, an extended structure "sHSVF" was defined, in which F is a "Validity Flag". Additionally, the structure "sValidityParameters" was defined. To make the code more clear, names of the fields in this structure were changed. The specific pixel now can be flagged as "IS VALID", "IS GRAY", "HAS DARK COLOR COMPONENT", "HAS OVERBLEACHED COLOR COMPONENT", "IS NEARLY GRAY". This flag may be instrumental for reliable color recognition and for reliable modification of the color of the pixels in accordance with the selected rules. By selecting values of the "sValidityParameters", the algorithm user can specify situations when the color of the specific pixel is set to the pre-defined value, marking problematic situations. The examples provided demonstrate that this approach can be used for reliable color recognition and advanced color manipulations for synthetic and real-life images**

Keywords-Image Processing; HSV; reliable color recognition; reliable color manipulations;.

I. INTRODUCTION

The standard inexpensive color digital camera on its output produces a sequence of bytes. In order to apply to this sequence basic imaging processing algorithms, this sequence is organized as a two-dimensional matrix of picture elements (pixels). Each pixel is a vector in the {R (red), G (green), B (blue)} space. In the inexpensive cameras, values of color components are in the range {0..255}. Presentation of the color as an {R, G, B} vector is quite natural for a human observer, having three types of day's vision color receptors. However, for applications used in machine vision, this presentation is not always practical. An alternative "traditional" presentation uses {H (hue), S (saturation) V (value)} space or {H, S, L (lightness)} space. Presentations of the pixels in the RGB, HSV, and HSL spaces are described in a number of classical image processing books [2]-[4]. Functions converting pixels in the RGB space to HSV and HSL spaces and back are well known and can be found in any programming language, including C-language. The value of H actually describes the pixel's color and thus can be used to recognize the pixel's color in a simple and convenient way.

There is a number of alternative approaches – for example, a sophisticated approach based on a sequence of different image processing algorithms designed for the specific goal [5]. A number of approaches to recognize fruits [6] and specifically apples [7] were tested. However, algorithms of that type use additional information about objects to be recognized and are in most cases too heavy for real-life applications.

Unfortunately, "traditional" plain and simple HSV and HSL algorithms have an inherent problem: Hue cannot be calculated if R, G, and B values are equal. When the presentation of pixel values in the range of byte {0..255] is used, "traditional" presentation HSV and HSL become problematic. One solution is to use the zero value of S (saturation) as a marker, pointing out that the value of H cannot be calculated in that case. However, in a practical situation, when an image has a noise (and images of digital cameras always have significant noise), the situation becomes even more problematic. It is clear that pixel {100,100,100} is gray, and Hue cannot be calculated in this situation. But if a digital camera produces a noise of, say 5 units, then pixel {98, 101, 104} must be treated as problematic for the reliable Hue calculation. Even though the value of Hue can be calculated in that case, it is clear that using this value for color recognition may lead to unreliable results.

Known properties of digital cameras' noise' require a rethinking of the way of Hue calculations in a number of additional problematic situations. For the synthetic image (an image created by software), an exemplary pixel {R=100, G=0, B=0} can be described as a pixel having pure RED color. However, for the digital camera that has 5 units of noise, those zero values are electronically problematic. The same is valid for the value equal to 255: this value, in most cases, means that the object is overbleached and that its color is distorted by clipping. Those and some other situations make the "traditional" HSV/HSL approach at least problematic for real-life applications.

The upper image in Figure 1 demonstrates a real-life photo of the mandarins. This image was taken with a typical smartphone camera. Nowadays, those cameras have powerful internal automatic image processing algorithms. In most cases, this results in an image optimized for a human observer. However, this, in many situations, results in color

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

distortion, which is "bearable" and even "preferable" for a human observer but is obviously a "no-go" for color recognition algorithms designed for a machine vision application. Callout P1a points to the pixel of the mandarin, the color of which is clearly different from other pixels of the same mandarin. Analyzing the profile of the line containing this pixel, one can see that the colors of the pixels in this region are clearly distorted: overbleached. Specific pixel marked as P1a on the image has a color (R=255, G=254, B=150) (see callout P1B). Value 255 for a real-life image must be considered as an "electronically clipped value" and must be treated as a "distorted value." Analyzing the pixel marked as P2a of the image, one can see that this pixel has a value {R=105, G=30, B=0} (see callout P2B on the lower profile). Value 0 for a real-life image must be considered as an "electronically clipped value" and treated as a "distorted value." Even though human observer probably recognize those pixels as "pixels of orange", machine vision must reject Hue's calculations of those pixels as non-reliable.

In order to better specify the problem with “traditional” Hue calculations, a special synthetic image was created. This synthetic Image (presented in Figure 2 - upper left image) has 6 strips {R, G, B} values of each strip are specified in the callouts. In order to demonstrate the challenge of reliable Hue calculations, strips have a slight green tint: the green component is “amplified” by adding values like 1 (in strip #1), 2 (in strip #2), etc. Additionally, low-level pseudo-random noise was added. Profile (presented in Figure 2 in the lower right image) and scaled-up fragment of strip #2 with saturation set to maximum (see the lower right image in Figure 2) demonstrate the effect of noise on the color of the specific pixel: when the color of the pixel is close to GRAY, the effect of noise can be significant (see strips #1, #2, and #3). It is clearly seen that while the original colors are a clear mixture of GREEN and GRAY, the colors in the lower right fragment in Figure 2 are actually pseudo-random. This effect of noise can be seen numerically in Figure 3. The upper image represents the Hue map of the original image after the addition of the noise. For strip #6 (the right part of the synthetic image), Hue value is about 85 – that is – mostly green (see an explanation of the Hue values in the Hue map later in the text). However, for strips #1, #2, and #3 (left parts of the synthetic image), the Hue value jumps unpredictably from 0 to 250, which means that the color in those strips is changed unpredictably, making the resulting values of Hue in this regions at least unreliable.

Again, even if the human observer probably recognizes pixels of this synthetic image as GREEN (because the human brain definitely can effectively eliminate noise in many practically important situations, machine vision algorithms must reject Hue's calculations of those noised pixels as non-reliable and mark them as “problematic”.

Earlier attempts to improve the HSV/HSL algorithm were described in [8] and [9]. The last published version is described in [1]. This article describes a more elaborate approach based on a number of additional modifications of the previously published versions in an attempt to provide a simpler algorithm that enables the implementation of reliable color recognition and reliable color manipulation.

As in the previous versions, an extended structure, “sHSVF,” is used, in which F stands for a "validityFlag".

Based on the previous experiments, the structure "sValidityParameters" was modified. Values of the fields in this structure make it possible to classify the specific pixel as "IS VALID", “IS GRAY”, "HAS DARK COLOR COMPONENT", “HAS OVERBLEACHED COLOR COMPONENT", "IS NEARLY GRAY", and properly set the "validityFlag" for each pixel of the image. By selecting values of the “sValidityParameters”, the user of the modified HSV algorithm can specify situations when the gray level of the specific pixel in the H-map is changed to the pre-defined color value, clearly marking problematic situations.

Section II describes the definitions of “sHSVF” structure (subsection ‘A’), “sValidityParameter” (subsection ‘B’), and flags used in specific situations (subsection ‘C’).

Section III presents changes in the “traditional” HSV algorithm.

Section IV presents exemplary analyses and processing of synthetic and real-life images demonstrating the properties of a modified algorithm.

Section V shortly summarizes the results obtained.

II. STRUCTURES SHSVF, SVALIDITYPARAMETER AND FLAGS

To store {R, G, B} values of the pixel, standard “sRGB” structure was used without changes:

```
struct sRGB
{
        unsigned char r;
        unsigned char g;
        unsigned char b;
    };
```

Standard “sHSV" structure was modified by using the “double” type and by adding the integer “valifityFlag”.

A. *Structure “sHSVf”*

The resulting “sHSVf” structure was defined as:

```
 struct sHSVf
{
        double H;   // Hue
        double S;   // Saturation
        double V;   // Value
        int    validityFlag; // Validity flag
    }; .
```

B. *Structure “sValidityParameter”*

Structure “sValidityParameter” was designed to set numerical values needed to mark problematic pixels. It was defined as:

```
 struct sValidityParameter
{
        double colorComponentMinValue;
        double colorComponentMaxValue;
        double saturationMinValue;
};
```

Usage of this structure will be described later.

C. Definitions of FLAGS

In order to properly mark different situations, a number of FLAGS were defined.

```
#define IS_VALID (0)
```

This flag is set when the value of HUE can be calculated. Traditionally, the Hue value is in the range {0..359}. However, in order to present Hue values as a gray image on the PC monitor, a scaling factor of 255/360 is applied. Then, the pixel in the Hue map is presented as a gray pixel having a value in the range {0..255}. For example, for the GREEN Hue, the "traditional" value is 120. However, in the Hue map (after applying the factor 255/360), the resulting value is 85. For the saturation map, value 0 means that the pixel is gray, whereas value 255 means that this pixel has a pure color (maximal saturation).

The following flags are used to mark a problematic situation in which Hue cannot be calculated, or this calculation can be considered as non-reliable. All problematic pixels are marked as COLOR (and not gray) pixels so that, as a human operator, as a machine vision algorithm can easily discard problematic pixels from a recognition process.

```
#define IS_GRAY (1)
```

In the classical HSV algorithm, DELTA of the specific pixel is calculated as a difference between the maximal and minimal values of the {R, G, B} values of the specific pixel. If DELTA is ZERO, this means that R=G=B and that this pixel is gray. In this case, Hue value cannot be calculated. Hence, this flag is set when DELTA of the specific pixel is ZERO. In this situation, the problematic pixel in the Hue map and in the Saturation map has MAGENTA color.

```
#define HAS_DARK_COLOR_COMPONENT (2)
```

This flag is set when at least one of the {R, G, B} values of the specific pixel is lower than the value specified in the parameter "colorComponentMinValue" in the structure "sValidityParameter". This situation is electronically problematic, hence those pixels of the Hue and Saturation maps are marked by a RED color.

```
#define
HAS_OVERBLEACHED_COLOR_COMPONENT (3)
```

This flag is set when at least one of the {R, G, B} values of the specific pixel is higher than the value specified in the parameter "colorComponentMaxValue" in the structure "sValidityParameter". This situation is electronically problematic hence those pixels of the HUE and Saturation maps are marked by a YELLOW color.

```
#define IS_NEARLY_GRAY  (4)
```

This flag is set if the calculated saturation value is lower than the value of "saturationMinValue" in the structure "sValidityParameter." A GREEN color marks those pixels on the Hue and Saturation maps.

Again, if none of the above flags were set, "sValidityParameter" would be set to the IS_VALID value (defined as zero). In this case, pixels in the Hue and Saturation maps are gray pixels, whereas the level of gray mapping Hue and Saturation values to the range of [0..255]. It must be noted that historically, in the Windows OS, values of Hue and Saturation were mapped in the {0..239} range. Some authors mapped values of Hue in the {0..360} range; however, this range cannot be presented in the standard displays designed for humans. Hence, the range [0..255] is better suited to the goal of this research.

III. CHANGES IN THE CLASSICAL HSV ALGORITHM

Classical function RGBtoHSV which is described at [2]-[4], and C-code of which is available in the public domain was modified by adding flags defined before. A complete code of the reworked function "ConvertRGBtoHSVf" is presented in Figure 4. This function is defined as:

```
void ConvertRGBtoHSVf(
        sRGB rgb,  sHSVf & hsvf,
        sValidityParameter param,
        int useLimits);
```

Arguments of the function are: "rgb" values of the current pixel as defined in the "sRGB" structure; values of the Hue, Saturation, Value, and Validity Flag of the above pixel to be calculated as defined in the "sHSVF" structure; "param" - specifying parameters used for the processing of this pixel; and flag "useLimits", which can be set to FALSE or TRUE. The value of the "V" can be calculated in any situation. When this flag is set to FALSE, values of Hue and Saturation are calculated in the "traditional way" without taking into account validity parameters (see lines 9-62 in Figure 4). Then, in the situation when Hue value cannot be calculated, values of Hue and Saturation are set to 0, which creates some well-known ambiguity: Hue=0 can mean RED color and/or GRAY pixel. When this flag is set to TRUE, {R, G, B} values of the pixel in the test are compared with the values specified in the "param" values (see lines 66-100 in Figure 4). A relevant color marks problematic pixels (see lines 76, in Figure 4), which enables to exclude them later from the following image analysis.

The reverse function ConvertHSVfToRGB is defined as:

```
void ConvertHSVfToRGB(
     sHSVf hsvf,
     sRGB& rgb);
```

This function has no significant differences from the "traditional" code.

As a reasonable self-test, a sequence of functions "ConvertRGBtoHSVf" - "ConvertHSVfToRGB" was run for a different sets of {R,G,B} values. No errors were found during those tests.

Additionally, functions converting source image to the Hue, Saturation, and Value maps, and to the image presenting "validityFlag" values as a human-readable gray map, in which different values are encoded by using different levels of gray were defined.

To demonstrate this approach to the well-known procedure of "recoloring", the exemplary function ChangeHue was defined as:

```
void ChangeHue(
        unsigned char trueColorSource[][..][..],
        unsigned char trueColorDestination[][..][..],
        double oldHue,  double newHue,
```

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

```
        double hueHalfRange,
        sValidityParameter param, int useLimits);
```

It must be noted that most recoloring algorithms replace the specified value of Hue with a new one. However, this approach is adequate only for synthetic images. The noise of the cameras "widens" the values of the selected Hue value. Hence, the additional parameter "hueHalfRange" is added to this function. Then, all pixels having valid values of Hue in the range from "(oldHue – hueHalfRange)" to the "(oldHue + hueHalfRange)" will be replaced with the "newHue" value. Naturally, this function uses validity parameters to exclude problematic pixels from processing. By setting values of those parameters, different image processing and color manipulation effects can be achieved.

IV. EXAMPLES OF ANALYSIS AND PROCESSING OF SYNTHETIC AND REAL-LIFE IMAGES

The upper image in Figure 5 presents a synthetic image specially prepared for the tests of a modified approach and its HSVF maps. This image contains 12 strips. The first strips are nearly gray (see callouts), whereas the last six have colors close to the six primary colors. The lower image presents profiles. Figure 6 demonstrates a color map of the upper image created by using a "traditional" algorithm (flag "useLimits" was set to "FALSE"). In this situation, three strips were marked as problematic: Hue cannot be calculated in those cases. Then, a low-level pseudorandom noise was applied to the image presented in Figure 6. The upper image in Figure 7 demonstrates profiles after the addition of the noise, whereas the lower image represents the Hue map. Hue recognition for most strips (except strips #4 and #5) is problematic – Hue calculations in those specially created problematic situations are non-reliable.

The upper left image in Figure 8 represents GREEN RAMP mixed with low-intensity RED RAMP. The lower left image represents the profile of this image. The upper right image represents the Hue map, and the lower right image is the profile of the Hue Map. It can be seen that Hue's calculations are not reliable for the "dark" regions. Hue values calculated for other parts of the ramp are noisy but reliable.

The upper left image in Figure 9 is a slightly problematic real-life photo of lemons and mandarins. The upper right image represents a Hue map calculated with the flag "useLimits" set to TRUE. Lower images represent corresponding profiles. Callout P1 points to the pixel having overbleached values for all color components; they are seen as pure white. They are marked by MAGENTA color, which is reserved for GRAY color (white in the HSV concept is gray because the saturation in this case is ZERO). Callout P2 points to the Hue value of a "typical" lemon. Callout P3 points to the pixel having a Hue value of the "typical" mandarin. The left image in Figure 10 represents parts of lemons as they were recognized by the modified algorithm. It can be seen that nearly all parts of lemons are reliably recognized and "recolored' to a strong blue color (except small overbleached regions marked by callout {253, 255, 254} and a small part of "too dark" lemons, which was not recolored). The right image demonstrates reliable recognition of mandarins.

Figure 11 represents another problematic image of mandarins. It can be seen that overbleached pixels are excluded from recognition.

V. SUMMARY AND CONCLUSIONS

The presented recognition and recoloring algorithm can be fine-tuned by setting dedicated for that goal parameters and flags. Examples given in Section IV demonstrated that the described approach could be used to analyze and process real-life photos. It is planned to rewrite the algorithm by using pointers and Q-numbers to improve the speed of the algorithm.

REFERENCES

[1] S. Kosolapov, "*Color Manipulation in Images by Using a Modified HSV Algorithm*", Think Mind, CONTENT 2023: The Fifteenth International Conference on Creative Content Technologies, Think Mind, pp. 1-6, 2023.

[2] J. Russ and F. Neal, "*The Image Processing Handbook*", CRC Press, 2017.

[3] R. Jain, R. Kasturi, and D. Schunck, "*Machine Vision*", MIT Press and McGraw Hill, Inc. 1995.

[4] J. Foley, A. van Dam, S. Feiner, and J. Hughes, "*Computer Graphics Principles and Practice*". Second Edition in C, ADDISON-WESLEY, 1997.

[5] J. Siyi and C. Heng, "*Color Recognition Algorithm Based on Color Mapping Knowledge for wooden Building Image", Scientific Programming*, Volume 2022, pp. 1-15, 2022.

[6] W.C. Seng and S.H. Mirisaee, "*A new method for fruits recognition system*", Electrical Engineering and Informatics, 2009. ICEEI '09. International Conference, Volume 01, 2009.

[7] W.Ji, D. Zhoo et al,, "*Automatic recognition vision system guided for apple harvesting robot*", Computers & Electrical Engineering, Volume 38, Issue 5, pp. 1186 – 1195, September 2012.

[8] S. Kosolapov, "*Comparison of Robust Color Recognition Algorithms*", Journal of International Scientific Publications, Materials, Methods and Technologies, Volume 15, pp. 274 – 283, 2021.

[9] S. Kosolapov, "*Evaluation of Robust Color Recognition Algorithms*", Journal of International Scientific Publications, Materials, Methods & Technologies, Volume 16, pp. 83-93, 2022.

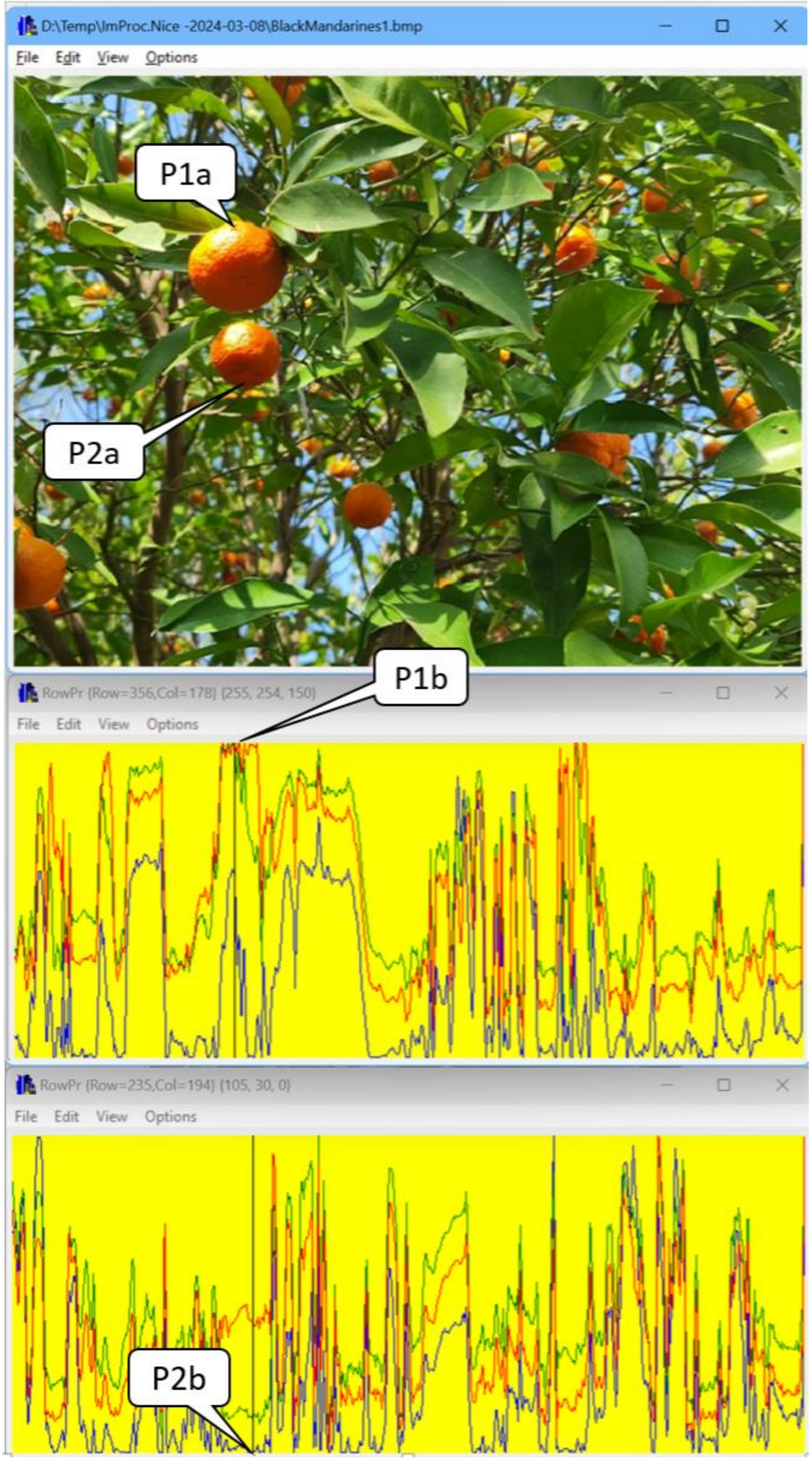

D:\Temp\ImProc.Nice -2024-03-08\BlackMandarines1.bmp
File Edit View Options
P1a
P2a
RowPr (Row=356,Col=178) (255, 254, 150)
P1b
File Edit View Options
RowPr (Row=235,Col=194) (105, 30, 0)
File Edit View Options
P2b

Figure 1. Upper image: real-life image of mandarines obtained by using typical smartphone camera (Samsung Galaxy Note 20). Image in the middle: profile for the line marked as P1a. The color of the pixel P1a is {R=255, G=245, B=150} (marked as P1b). It is clearly seen, that color of this pixel is distorted (overbleached). The Lower image is a profile of the line marked as P2a. The color of the pixel is {R=105, G=30, B=0} (marked as P2b). It can be stated that the color of this pixel is distorted (B component has a zero value).

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

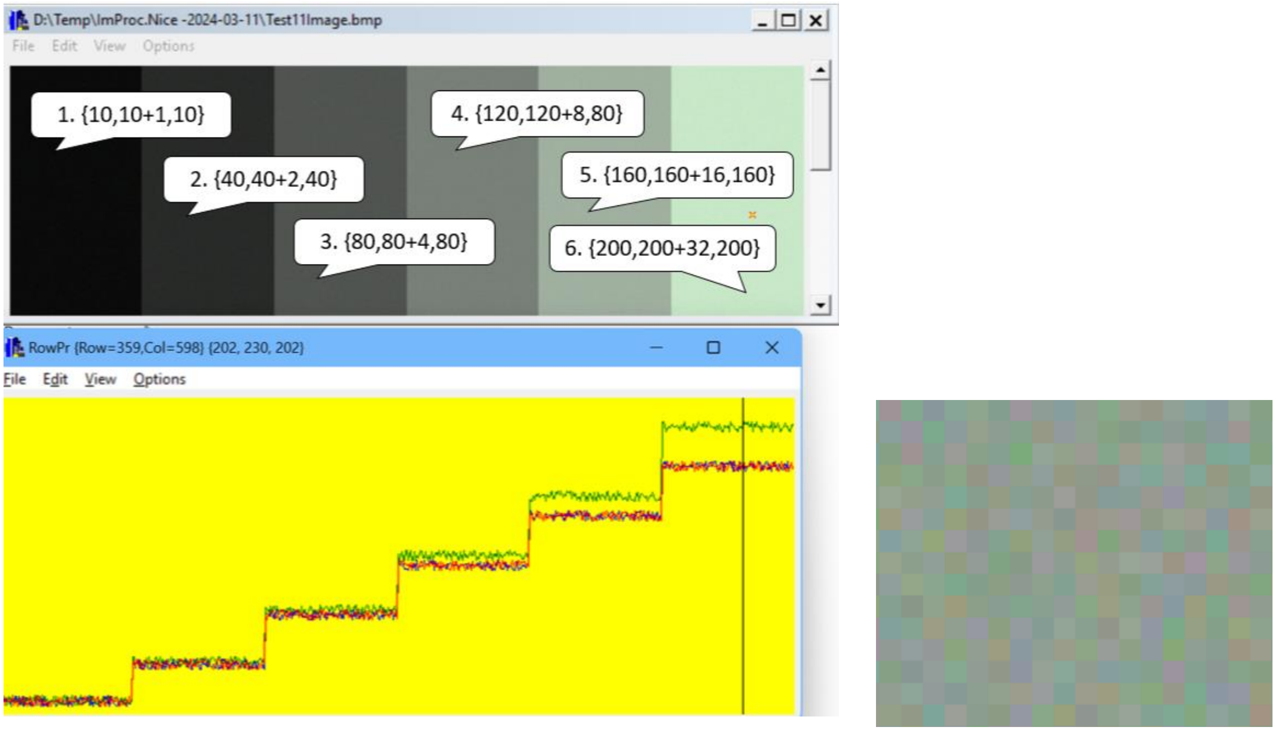


Figure 2. Upper Left: Synthetic image – 6 strips, having mix of GRAY + GREEN colors. (see callouts). Pseudo-random noise is added. Lower left: Profile of a row = 359. Specific values of the pixel of cursor are {R=202, G= 230, B = 202}, and because of noise values are slightly differ from declared values {R=200, G=232, B=200}. Lower right: scaled UP fragment from the strip #2. Saturation and brightness were set to maximum to emphasize that because of noise, nearly gray pixels (with some green tint) now have pseudorandom colors (red, green, blue, etc.).

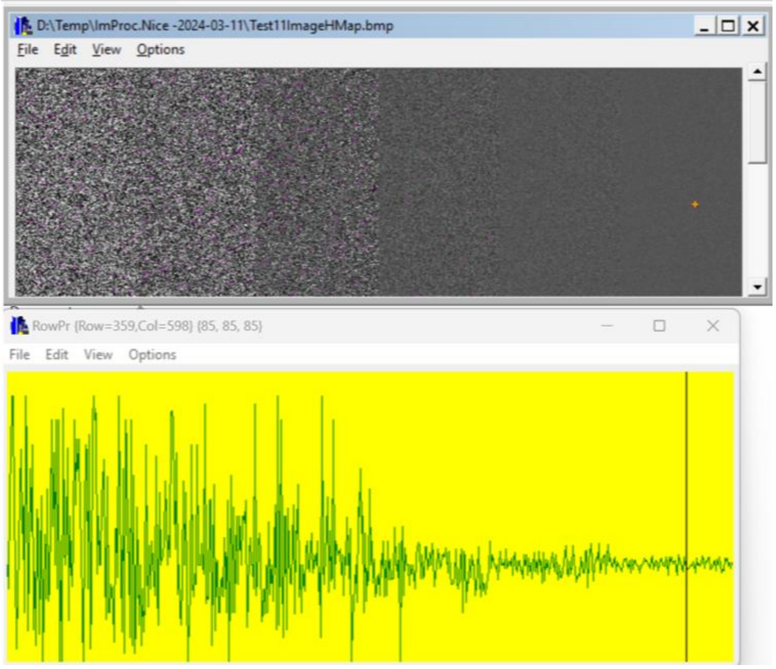
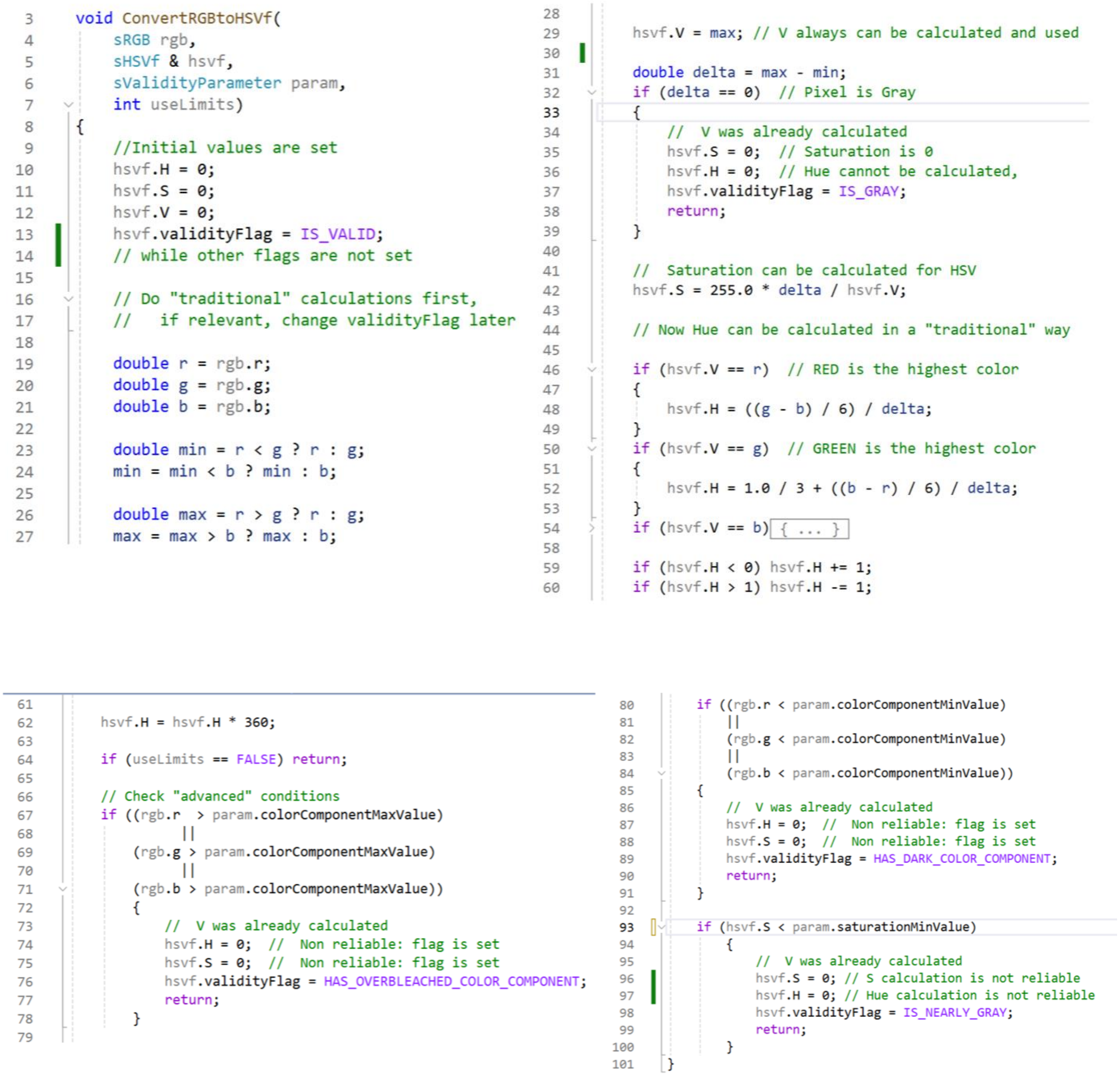

D:\Temp\ImProc.Nice -2024-03-11\Test11ImageHMap.bmp
File Edit View Options
RowPr {Row=359,Col=598} {85, 85, 85}
File Edit View Options

Figure 3. Upper image: Hue map calculated by a “traditional” algorithm. Lower image: profile of the upper image. It can be seen that the Hue value in strip #6 is about 85 (which corresponds to GREEN). Additionally, it can be seen that for lower intensities, Hue value is nearly pseudorandom – so that colors of the pixels are nearly pseudorandom – as it can seen in the lower right image in Figure 2.

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org



```
3   void ConvertRGBtoHSVf(
4       sRGB rgb,
5       sHSVf & hsvf,
6       sValidityParameter param,
7       int useLimits)
8   {
9       //Initial values are set
10      hsvf.H = 0;
11      hsvf.S = 0;
12      hsvf.V = 0;
13      hsvf.validityFlag = IS_VALID;
14      // while other flags are not set
15
16      // Do "traditional" calculations first,
17      //   if relevant, change validityFlag later
18
19      double r = rgb.r;
20      double g = rgb.g;
21      double b = rgb.b;
22
23      double min = r < g ? r : g;
24      min = min < b ? min : b;
25
26      double max = r > g ? r : g;
27      max = max > b ? max : b;
```

```
28
29      hsvf.V = max; // V always can be calculated and used
30
31      double delta = max - min;
32      if (delta == 0)  // Pixel is Gray
33      {
34          //  V was already calculated
35          hsvf.S = 0;  // Saturation is 0
36          hsvf.H = 0;  // Hue cannot be calculated,
37          hsvf.validityFlag = IS_GRAY;
38          return;
39      }
40
41      //  Saturation can be calculated for HSV
42      hsvf.S = 255.0 * delta / hsvf.V;
43
44      // Now Hue can be calculated in a "traditional" way
45
46      if (hsvf.V == r)  // RED is the highest color
47      {
48          hsvf.H = ((g - b) / 6) / delta;
49      }
50      if (hsvf.V == g)  // GREEN is the highest color
51      {
52          hsvf.H = 1.0 / 3 + ((b - r) / 6) / delta;
53      }
54      if (hsvf.V == b){ ... }
58
59      if (hsvf.H < 0) hsvf.H += 1;
60      if (hsvf.H > 1) hsvf.H -= 1;
```

```
61
62      hsvf.H = hsvf.H * 360;
63
64      if (useLimits == FALSE) return;
65
66      // Check "advanced" conditions
67      if ((rgb.r  > param.colorComponentMaxValue)
68                ||
69          (rgb.g > param.colorComponentMaxValue)
70                ||
71          (rgb.b > param.colorComponentMaxValue))
72          {
73              //  V was already calculated
74              hsvf.H = 0;  //  Non reliable: flag is set
75              hsvf.S = 0;  //  Non reliable: flag is set
76              hsvf.validityFlag = HAS_OVERBLEACHED_COLOR_COMPONENT;
77              return;
78          }
79
```

```
80      if ((rgb.r < param.colorComponentMinValue)
81          ||
82          (rgb.g < param.colorComponentMinValue)
83          ||
84          (rgb.b < param.colorComponentMinValue))
85      {
86          //  V was already calculated
87          hsvf.H = 0;  //  Non reliable: flag is set
88          hsvf.S = 0;  //  Non reliable: flag is set
89          hsvf.validityFlag = HAS_DARK_COLOR_COMPONENT;
90          return;
91      }
92
93      if (hsvf.S < param.saturationMinValue)
94          {
95              //  V was already calculated
96              hsvf.S = 0; // S calculation is not reliable
97              hsvf.H = 0; // Hue calculation is not reliable
98              hsvf.validityFlag = IS_NEARLY_GRAY;
99              return;
100         }
101 }
```

Figure 4. Full code of the modified function ConvertRGBtoHSVf. Code between lines 3-62 is a classical well-known code. Lines 54-99 contains proposed modifications. See additional explanations in the text.

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

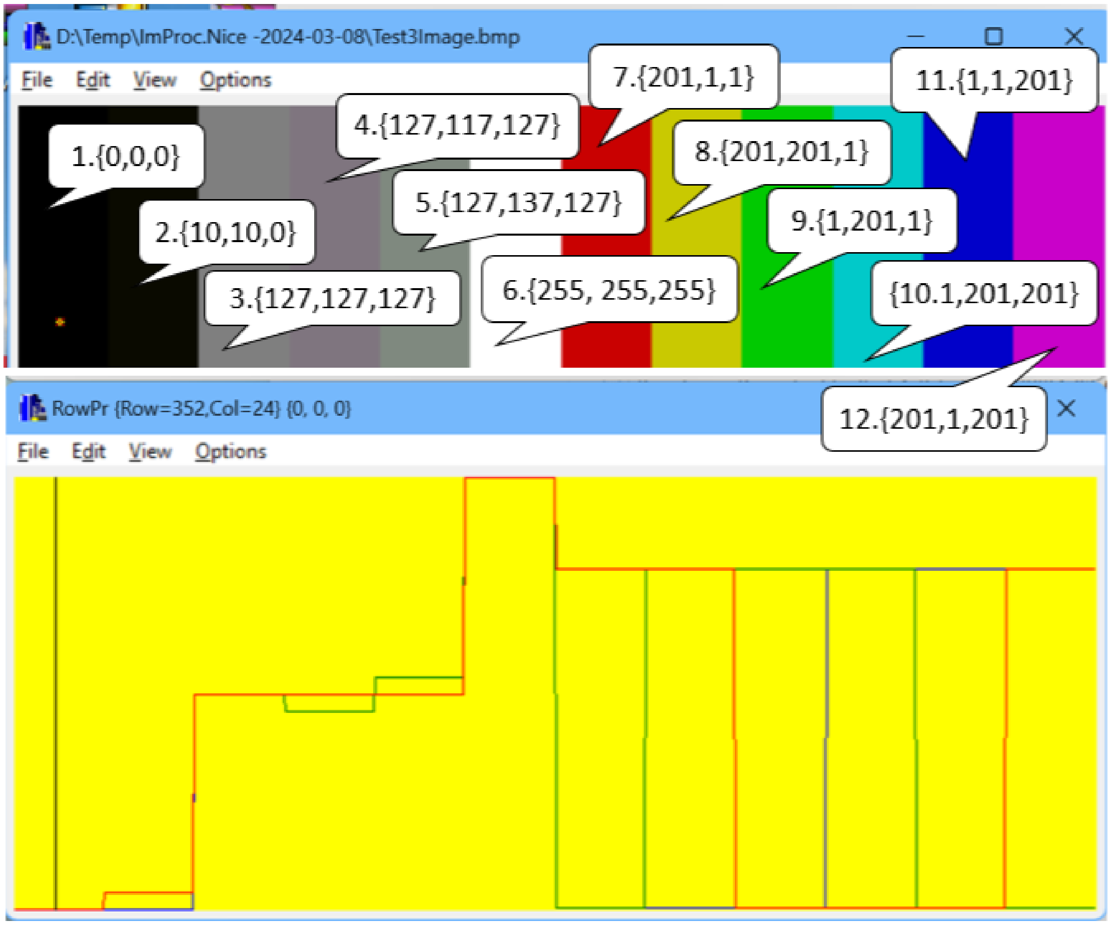


Figure 5. Upper Image: Synthetic Test Image containing 12 strips having selected {R, G, B} values. Callouts specified those values for every strip. Lower image: R, G and B profiles of the upper image. The strip #1 is totally black : {0,0,0}. Considering future tests, some 0 values were shifted to value 1, or to another value (see values in callouts). See additional explanations in the text.

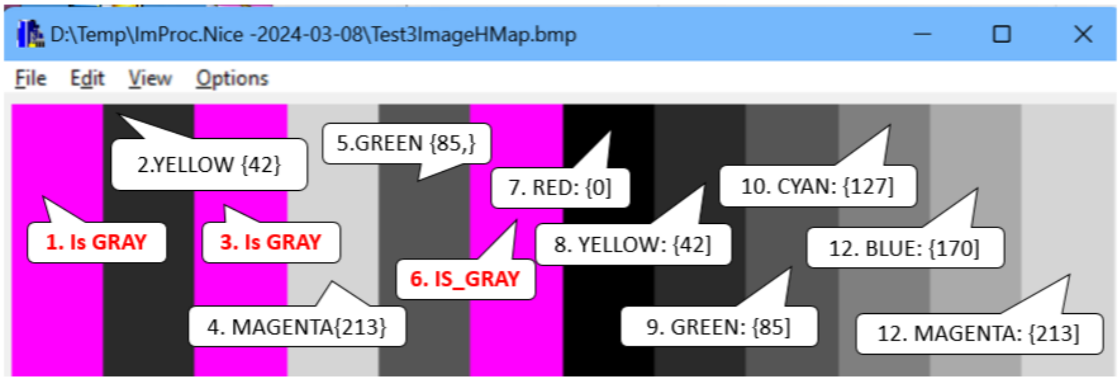


Figure 6. H-Map of the Test Image presented in Figure 5. Flag "useLimits" was set to "FALSE", hence the processing was provided by a "traditional" part of the function "ConvertRGBtoHSVf", without modifications. The strips #1, #3, and #6 were marked by the "traditional" algorithm by MAGENTA color because their original pixels are exact gray pixels (see callouts in Figure 5. In this case, pair of strips #2 and #8 has the same Hue =42 (corresponding to a YELLOW color), despite the fact that most human observers will not be able to detect color of the very dark strip #2. The same is true for strips #5 and #9 (Hue = 85, which corresponds to the GREEN color of the strips). In this case, the human observer can validate that strip #5 has a GREEN tone. Original {R,G,B} values {127, 137, 127} (see callout 5 in Figure 5) confirmed that this color is a mix of gray {127, 127, 127} with dark GREEN {0,10,10}.

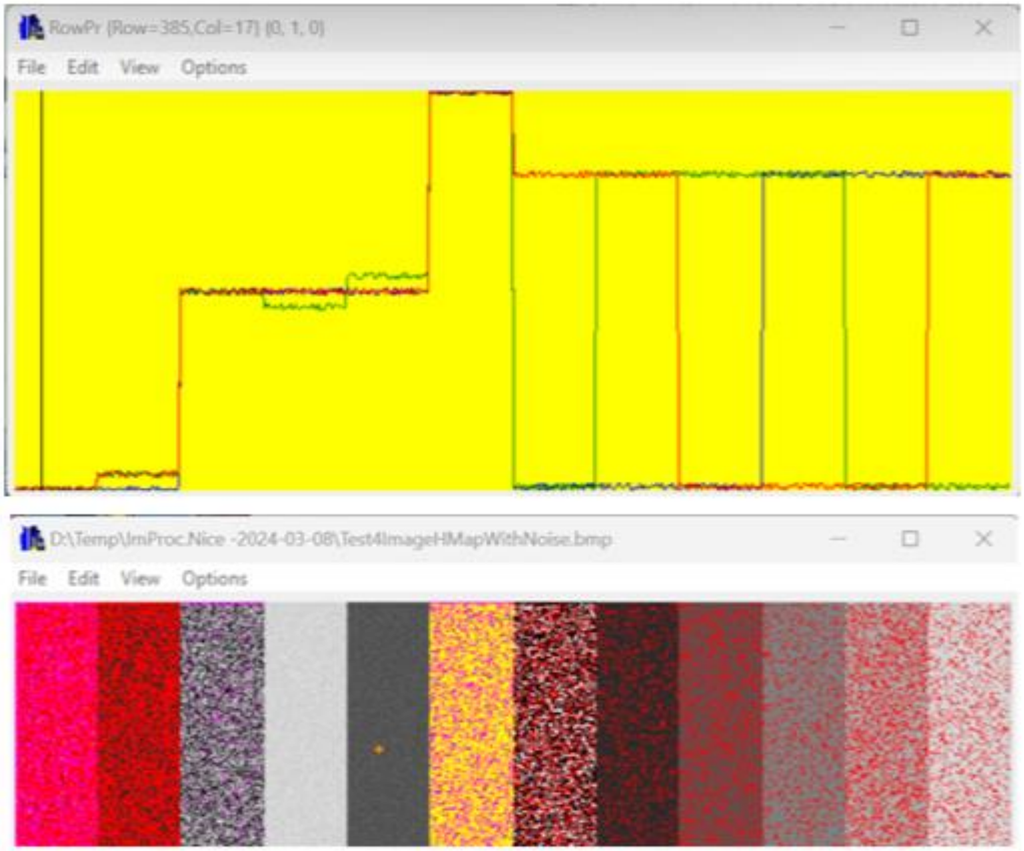

RowPr (Row=385,Col=17) (0, 1, 0)
File Edit View Options
D:\Temp\ImProc.Nice -2024-03-08\Test4ImageHMapWithNoise.bmp
File Edit View Options

Figure 7. Pseudo-random noise was added to the Test Image from the Figure 5. This pseudo-random noise can be clearly seen in the Upper Image – profiler of row 385. Lower image: Flag “useLimits” was set to “TRUE”; hence the processing was provided with a “modified” part of the function ConvertRGBtoHSVf. It is clear that even small level of noise makes color recognition for the selected values problematic.

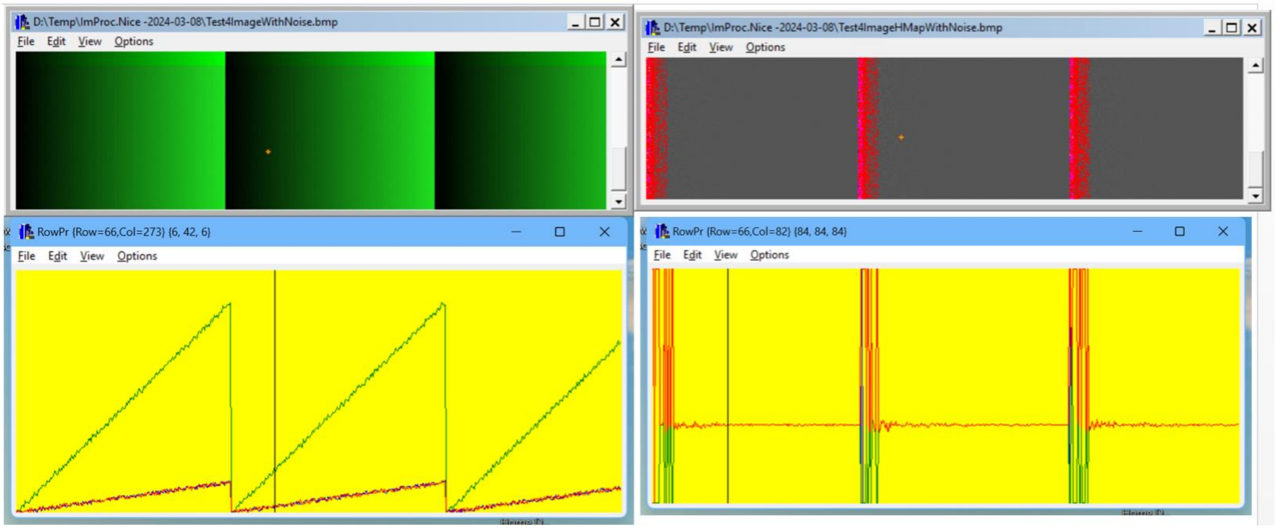

D:\Temp\ImProc.Nice -2024-03-08\Test4ImageWithNoise.bmp
File Edit View Options
D:\Temp\ImProc.Nice -2024-03-08\Test4ImageHMapWithNoise.bmp
File Edit View Options
RowPr (Row=66,Col=273) (6, 42, 6)
File Edit View Options
RowPr (Row=66,Col=82) (84, 84, 84)
File Edit View Options

Figure 8. Upper left: Synthetic image GREEN RAMP mixed with low-ntensity RED RAMP. A low-level pseudo-random noise was added. This noise can be seen in the Lower Left image: profile of the Upper left image. Upper right image: Hue map of the Upper Left image. It can be seen that for most pixels reliable color recognition is possible (Hue value is about 85 which corresponds to GREEN – see Lower Right profile). However, values of Hue for dark” regions of this ramp are marked by RED color (signalling that at least some values of some color components are too low).

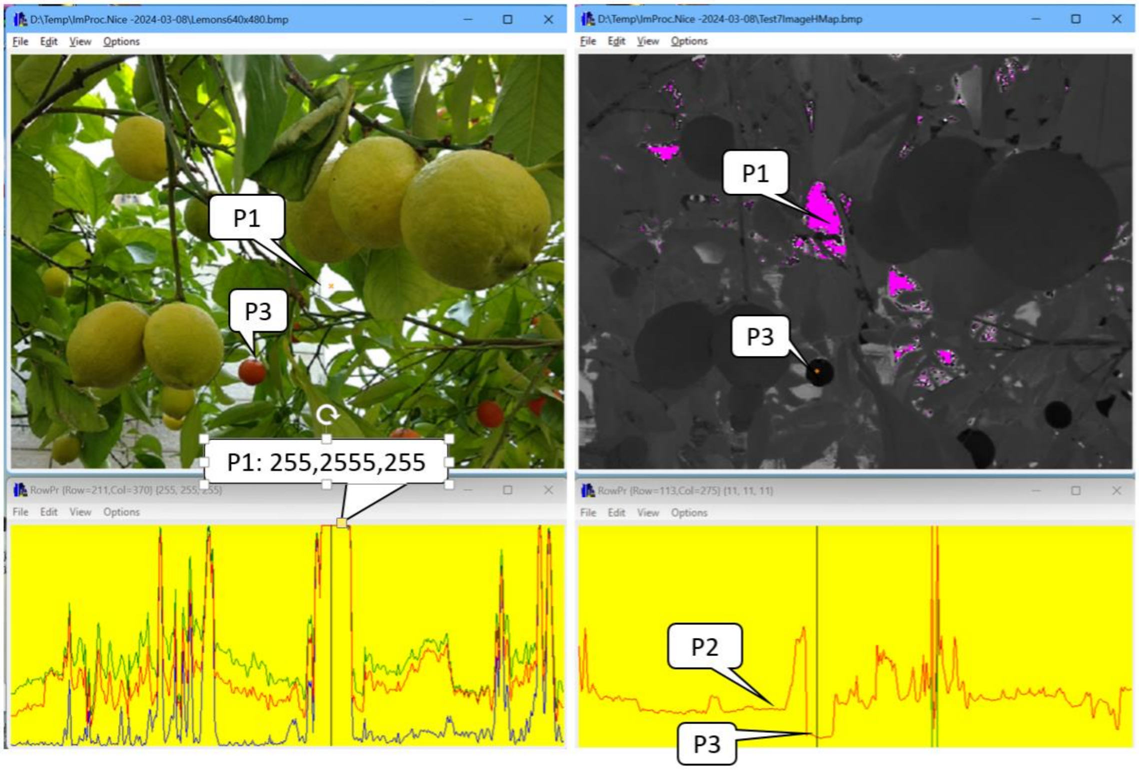

D:\Temp\ImProc.Nice -2024-03-08\Lemons640x480.bmp
File Edit View Options
D:\Temp\ImProc.Nice -2024-03-08\Test7ImageHMap.bmp
File Edit View Options
P1
P3
P1
P3
P1: 255,2555,255
RowPr (Row=211,Col=370) {255, 255, 255}
File Edit View Options
RowPr (Row=113,Col=275) {11, 11, 11}
File Edit View Options
P2
P3

Figure 9. Upper left: Real-life image of lemons and mandarines. Upper right: H-Map of the Upper left image. Lower right: profile of the line of pixel marked as P1. Lower right: profile of row marked as P3. See the discussion in the text.

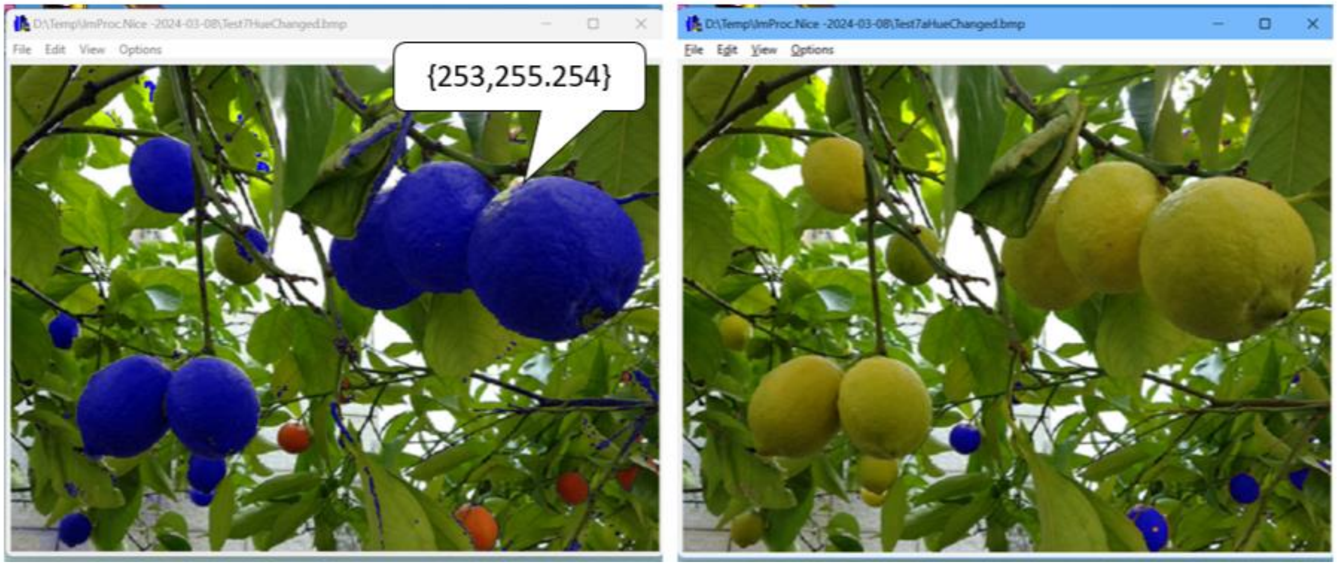

D:\Temp\ImProc.Nice -2024-03-08\Test7HueChanged.bmp
File Edit View Options
{253,255.254}
D:\Temp\ImProc.Nice -2024-03-08\Test7aHueChanged.bmp
File Edit View Options

Figure 10. Left image: Recognition of lemons from the upper left image in Figure 9. Recognized parts of lemons are recolored to the bright BLUE color. Non-recognized pixels retain their original colors. Right Image: Recognition of mandarins from the upper left image in Figure 9. Recognized parts of mandarines are recolored to the bright BLUE color. Non-recognized pixels retain their original color. See additonal details in the text.

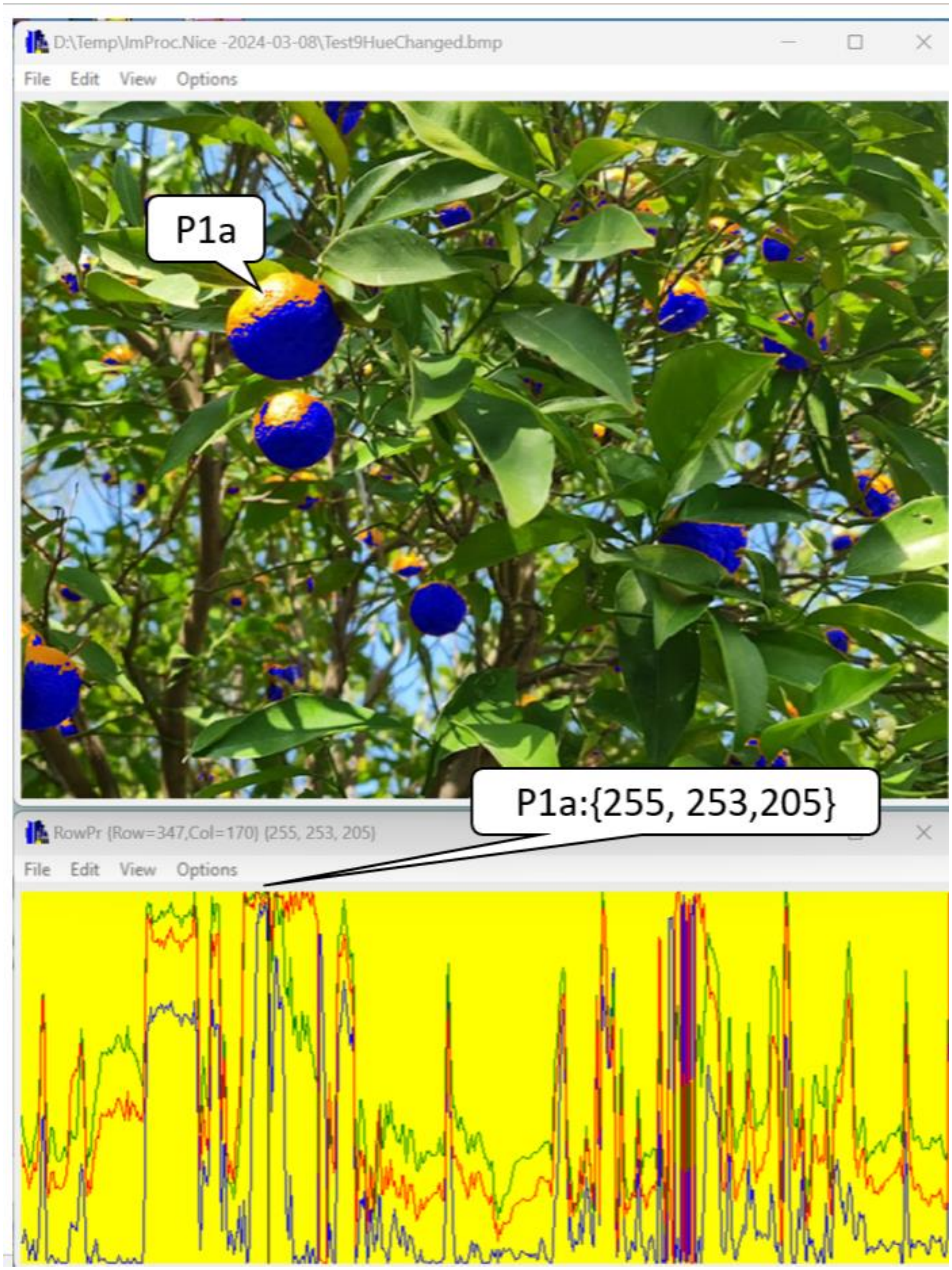

D:\Temp\ImProc.Nice -2024-03-08\Test9HueChanged.bmp
File Edit View Options
P1a
P1a:{255, 253,205}
RowPr (Row=347,Col=170) (255, 253, 205)
File Edit View Options

Figure 11. Upper image: results of recognition of the mandarines from the image presented in Figure 1. Lower image: profile of the line marked as P1a. Pixels near P1a have an obvousely distorted color {255, 253. 205}. Experienced human operator without hesitations would mark those pixels as “part of the mandarine”; however, in the frames of the discussed approach, they must not be recognized as having reliable Hue value, and, thus, those pixels were not recolored.

The Project Management Assistant that Never Sleeps

Using Generative Artificial Intelligence to Support Project Management Activities

Claudia Hess and Sibylle Kunz
IT & Engineering
IU International University of Applied Sciences
Erfurt, Germany
e-mail: claudia.hess@iu.org; e-mail: sibylle.kunz@iu.org

***Abstract*—Over the past few years, people have become accustomed to using a variety of digital tools in their work environment, e.g., for collaboration, knowledge management and task management. Recently, people have become aware of a new type of digital tool: Generative Artificial Intelligence (AI) assistants, such as ChatGPT, Gemini or Copilot, which are capable of generating text as well as other forms of content, such as images, music, and even code in response to user queries in natural language. These tools have a huge impact on the way we work. New application scenarios are constantly being explored and Generative AI assistants are being used in fields such as marketing, medicine, and education. This paper discusses a use case in the field of project management. It analyzes how Generative AI assistants could support project management tasks performed by project managers and project management assistants. It also discusses the new skills that these persons must acquire in order to interact with the tools in an efficient way and to use the results provided responsibly. An extensive case study shows how selected project management tasks, i.e., stakeholder management, can benefit from the support of a Generative AI assistant. The case study covers a wide range of project-related activities, including its use in brainstorming, creating a stakeholder matrix, writing meeting invitations, and sentiment analysis. The results of the GenAI assistant for each task are evaluated while special attention is given to the aspects of trust and control in such settings.**

Keywords-Large Language Model (LLM); generative Artificial Intelligence; project management; stakeholder management; prompt engineering.

I. INTRODUCTION

As in almost every area of our personal and professional lives, tools based on generative artificial intelligence (AI) have been explored in the field of project management over the past year. There is a wide range of application scenarios in project management and the need for project managers and project management assistants to become proficient users of these tools to increase not only their productivity but also their creativity [1]. Project management professionals already use a variety of digital tools in their daily work, ranging from project planning, and monitoring to stakeholder and risk management [2]. Some of these tools are highly specialized, such as project management software, while others are not limited to project management, such as video conferencing software, instant messaging applications, cloud-based document sharing, or digital whiteboards. These digital tools have fundamentally changed the way people work. Recently, a new technology has emerged that has the potential to have a profound impact on modern work culture, including project management: Generative AI (GenAI), which is able to create different types of content, such as text, images, music and videos. GenAI assistants provide an easy-to-use interface to this technology. They simulate human-like conversations and provide meaningful responses to user queries [3]. Due to recent advances in AI, these AI assistants are becoming increasingly powerful.

As the use of digital tools is already widespread in project management, it can be assumed that many project managers will use generative AI in their daily work. They are likely to adapt these new virtual assistants easily and quickly [4]. Various blog posts, and "cheat sheets" showcase examples of how to interact with these tools [5]. However, there are many differences in the way of interacting and interpreting the results obtained by an GenAI assistant in comparison with classical tools in project management. The project manager and the project team should be aware of them and possess certain skills to use them in a productive way. Moreover, despite all the new and exciting possibilities that GenAI offers, there are risks involved that need to be mitigated, such as ethical challenges like discrimination, information hazards, misinformation harms and human-computer interaction harms [6], [7].

To this end, the paper is structured as follows. Section II provides the required theoretical foundation on GenAI and the new skill "prompt engineering". Section III discusses how project management professionals can benefit from GenAI assistants in their daily work. Therefore, it structures AI-assisted project management tasks in six categories and analyzes the required skills. As a concrete case study, Section IV demonstrates how a GenAI assistant can support the activities around stakeholder management. Useful prompts will be introduced. Recommendations will be given on how to fine-tune these prompts to improve the quality of responses and their appropriateness to the particular project context. Based on the experiences with the case study, Section V discusses potential and risks of the use of generative AI in project management. Section VI concludes the paper and outlines areas for future research.

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

II. GENERATIVE AI AND PROMPT ENGINEERING

A. *Generative AI*

Since OpenAI released ChatGPT to the public in November 2022, GenAI gained much attention. The term GenAI denotes a subfield of AI that specializes in developing models that are capable to create new, original content [8]. Based on the resulting media, we can differentiate four domains of GenAI: text generation, image generation, audio generation and video generation [8]. Each of it uses specific models and algorithms to generate the desired output based on learned patterns.

Large Language Models (LLMs) are the common architecture for text generation [9]. LLMs such as GPT4 from OpenAI have been trained on vast amounts of text data from the Internet. Their development has been made possible by the recent advances in AI, especially deep learning and transformer architectures, the rapid growth in computational power and the large amount of data available on the Internet that could be used as training data [10]. LLMs have learnt based on this extensive training set to generate text sequences that most likely continue a given text [11]. Nowadays, the quality of the text produced by these systems is so high that it is difficult to distinguish whether a text was written by an AI or by a human [12].

In image generation, diffusion models are commonly used for text-to-image generation [9]. Tools such as DALL-E offer the possibility to create new images based on textual descriptions or to modify existing images in novel ways. Audio generation employs models to synthesize speech, music, or other sound effects that can mimic real-world audio. Video generation combines techniques from both image and audio domains to create or modify video content.

End users nowadays benefit more and more from models that take different types of input and are capable of generating output in various forms. These models are referred to as multimodal [9]. There are user-friendly interfaces that allow end users to benefit from the capabilities of such GenAI models without the need for advanced AI or programming knowledge. Examples are ChatGPT from OpenAI, Google's Gemini or Microsoft's Copilot. Users can formulate their input in natural language (or even provide images) and, as we know from chatbots, these models generate coherent answers, simulating a conversation between human beings [3]. By providing easy-to-use interfaces, the model-specific capabilities are extended so that these models can be used in a variety of real-world use cases [9]. We refer to these tools in the following as GenAI assistants.

It is important to know the differences between LLMs and search engines. They are distinct technologies, as they are characterized by different capabilities. LLMs generate context-sensitive text based on a given input, formulated in natural language [3]. In contrast, search engines take a keyword-based query as input and retrieve information from indexed web pages or databases. They do not generate new content but provide access to existing information. This is not the case with LLMs, which might "hallucinate" facts and references [3]. Moreover, LLMs cannot provide information on recent events without being integrated into other tools that have access to up-to-date information (such as search engines) [3]. Users must be aware of these limitations.

B. *Prompt Engineering*

The instructions given to a LLM are called prompts. In the prompt, the user provides the LLM with the context of the conversation and defines how the intended output should look like in terms of content and format [13]. For instance, a user might specify the desired output as an essay with approximately 1500 words. The process of constructing these prompts is referred to as *prompt engineering*, i.e., "the art of fine-tuning the questions or commands provided to AI models in order to optimize their performance and guarantee that they produce the desired results" [14]. In consequence, to collaborate efficiently with LLMs, users need to have a good knowledge of prompt engineering [3].

III. GENAI ASSISTANTS IN PROJECT MANAGEMENT

A. *Usage of GenAI in Project Management and Project-Related Tasks*

GenAI can be applied to the discipline of project management in a variety of ways in order to assist typical activities such as project planning [4], [15], stakeholder management [1] or risk management [4]. An example is the use of GenAI in project estimation [16]. As result to a first prompt, a list of possible tasks is created as Work Breakdown Structure (WBS). For each WBS task, the required time was then added manually. Then, the extended list was handed over to the GenAI together with a list of available resources (such as the number of developers) and their average salary. Based on this information, the GenAI calculated the total effort for the project. In this very simple example, the calculations made by the GenAI were correct. However, project managers should be aware that LLMs might have problems with solving more complex mathematical tasks [17].

Further examples of how GenAI can assist tasks performed within project teams include idea generation [18], assistance in requirements gathering [16], digital prototyping [18], and coding [19]. These use cases indicate a shift in the perception and application of AI. Traditionally associated with more analytical tasks, AI, and particularly GenAI, is now being used more and more for creative tasks.

To assess whether a GenAI assistant's results are really helpful, or even as impressive as some AI enthusiasts claim [20], Barcaui and Monat conducted a qualitative study comparing the content as well as the structure of a project plan created by an experienced project manager and a GenAI [15]. The human project manager and the GenAI were provided with the same information about a planned beauty app and given the task to plan the project. They had to define the scope, develop the timeline, estimate costs, assess resources, and analyze stakeholders, etc. Barcaui and Monat compared a large amount of data generated by the human project manager in about three weeks and by the GenAI assistant in 15 rounds with elaborated prompts. In order to benefit from the strengths of both humans and GenAI, they recommend a collaborative approach. This is also recommended in this paper. GenAI can produce remarkable results very quickly, but it may lack the necessary domain knowledge and understanding of the particular environment, in which the project is taking place. Therefore, it is important that human project managers enrich the prompts with context and carefully review the results. As discussed in the previous section, excellent prompt engineering skills are required.

B. Patterns of AI-Assisted Project Management Tasks

In the following, six patterns are introduced that represent “meta-tasks” performed by project managers within typical project management activities such as project planning, project estimation, stakeholder management or risk management. Figure 1 provides an overview of the defined patterns, sorting them by project duration and showing the degree of interaction between humans and GenAI assistants. For greater clarity, each pattern is only shown for the primary area of application. It goes without saying that patterns can also be used repeatedly during the course of a project. The intensity of the interaction can vary widely, ranging from a basic information exchange to collaborative partnerships where humans and GenAI assistants work closely together on creative or problem-solving tasks.

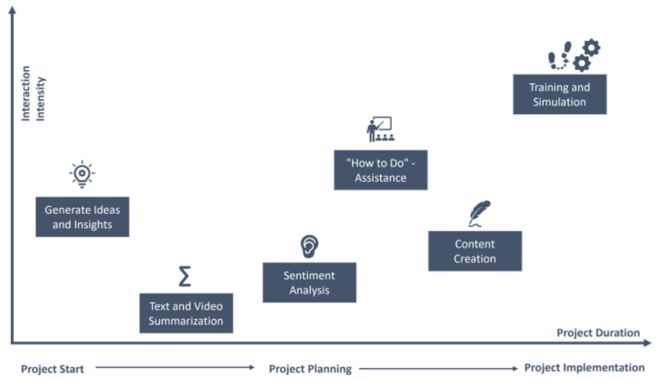


Figure 1. Patterns of AI-Assisted Project Management Tasks

Given the rapid advances in generative AI, the following overview is not meant to be exhaustive. Rather, it aims to provide a framework to structure practical AI-assisted project management tasks.

1) Generate Ideas and Insights: A GenAI assistant can support idea generation and brainstorming by suggesting topics or themes based on user input or some predefined categories. Combined with search engines, they can retrieve information from a variety of sources, such as articles, research papers, or databases. However, users need to check the output generated by the GenAI assistant for their correctness [3], as a high likelihood of the generated content is no guarantee that it is correct [6]. A project manager could, for example, ask a GenAI assistant to generate a list of risks or ideas for risk mitigation strategies.

2) “How to Do”- Assistance: Users can ask a GenAI assistant for guidance on a certain task, for instance, how to write a project proposal or how to plan and estimate the project. The GenAI assistant typically provides a list of steps for the user to follow, often like a recipe, e.g., certain aspects that need to be clarified for the project proposal. It should be noted that, if the prompt is too general, the result will also be unclear or inaccurate [21].

3) Text and Video Summarization: GenAI assistants are able to summarize or extract key aspects of large texts. This ability could save time for a project manager who needs to compile and summarize project data for a status update. Instead of reviewing numerous documents and reports, they could focus on interpreting the results. However, this task requires a significant level of trust in the GenAI assistant. New application scenarios came up with the integration of GenAI in video conferencing systems. GenAI can be employed to listen to or read through transcripts of discussions and automatically generate summaries or highlight critical aspects. For instance, Microsoft included a feature called Intelligent Recap into Microsoft Teams, which allows you to create meeting minutes including follow-up activities [22].

4) Content Creation: GenAI assistants can support in generating content for various purposes, such as articles, blog posts, social media updates, or product descriptions [3]. Based on a given topic, they can, for example, offer relevant keywords, suggest headlines, or write complete articles. In project management, different types of content are relevant, such as official documentation like project proposals, one-pagers, reports, internal working documents, as well as any communication with stakeholders. In addition to generating new texts, GenAI assistants can improve existing text. On the one hand, it can correct a text in terms of spelling, punctuation, and grammar. This is especially helpful for non-native speakers. On the other hand, it can rewrite a text and change its writing style based on the desired tone and target audience. For instance, it could turn a formal and technical text into a conversational and engaging one. This could be useful when preparing two communications about a milestone achieved in the project, one for the project steering committee, and the other for future users.

In addition to generating text, advanced GenAI assistants now have the ability to generate images and other forms of media. These systems are able to process and interpret information not only in the form of text, but also in multiple modalities, including images, audio, and video. In project management, for example, a GenAI assistant could create visualizations of project data, such as graphs or charts, based on text prompts. A project manager preparing a presentation for the steering committee could use the virtual assistant to create visualizations of the project's progress.

5) Training and Simulation: A GenAI assistant can help users explore different perspectives by simulating to be someone else. By pretending to be a particular persona (“role prompting”) with certain opinions, experiences, or expertise, the chatbot can provide responses from that particular perspective. This allows users to gain insights that they might not have considered otherwise. In project management, this could be useful in stakeholder management. A variation would be a flipped interaction, where the GenAI assistant asks questions and the user has to answer. This could be used to train difficult situations.

6) Sentiment Analysis: GenAI assistants can identify sentiments or emotional tones expressed in a given text [23]. They can assist users in gaining insights into the overall sentiment or mood of the content. This capability could be used in stakeholder management, for instance, to analyze comments posted by stakeholders on social media or other online communications.

C. Required Skills

To interact successfully and effectively with GenAI assistants, humans need certain skills. With the previously described tasks in mind, these skills are explored below. Figure 2 provides an overview of these skills.

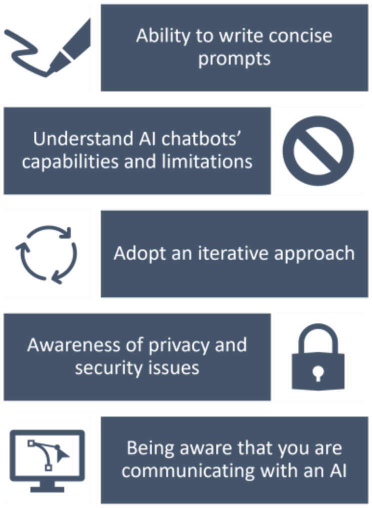

Ability to write concise prompts
Understand AI chatbots' capabilities and limitations
Adopt an iterative approach
Awareness of privacy and security issues
Being aware that you are communicating with an AI

Figure 2. Skills required for efficient collaboartion with GenAI.

1) Ability to Write Concise Prompts

When writing prompts, users should ask specific questions and be concise in their wording, as this helps chatbots to better understand user queries [24]. In addition to the use of simple language, the questions, as well as the provided content should be clearly structured [14]. By learning to provide GenAI systems with clear, unambiguous prompts, users can guide the model to produce more accurate, relevant, and useful outputs. This skill involves not only understanding what to ask, but also how to ask it in a way that matches the AI's processing capabilities and limitations. Various guidelines and learning paths on prompt engineering can be found online. It should be noted, however, that these were typically compiled for a certain tool in a certain version, e.g., GPT-4. Because of the rapid development in the field of GenAI, prompts that that yielded excellent results with an earlier version of a model might not work as effectively with a newer version, which may have been trained on more data, utilize different algorithms, or have altered processing mechanisms. Moreover, a prompt that works well for one tool may not give good results for another tool. This might also be the case for all prompts discussed in the following. However, some basic prompting techniques that will be applied within the case study are presented.

a) Use multi-turn prompting: First, users should think about whether they want to integrate several tasks into a single prompt or to split them into multiple prompts. For instance, a GenAI assistant could be asked in a single prompt to generate four ideas about a certain topic and to discuss them in a short blog post. Alternatively, a first prompt could ask for four ideas, and a second one to write a blog post based on idea 1, idea 2, and so on. The advantage of using multiple prompts is that the output can be reviewed, and if necessary, adjusted or extended. This technique, also referred to as “chain of thought prompting” or “multi-turn prompting” takes into account the potential complexity of certain tasks and aims to facilitate more interactive and collaborative interactions with the GenAI [25], [26].

b) Provide context information: A second important technique is to incorporate context information, such as background knowledge, domain-specific knowledge, or insights about the target audience, into the prompt. Using such knowledge-augmented queries supports the LLM to generate more informed and accurate responses, because the model gains a more complete understanding of the specific requirements, nuances, and the users' expectations [21], [27]. However, users should be careful about providing sensitive or private information.

c) Provide behavioral instructions: In addition to providing context information, it is a good practice to add some behavioral instructions, which frame the conversation with the GenAI assistant and offers a clear direction for the AI in terms of tone, style, and content. Including behavioral instructions serves as a valuable starting point for prompts. An example is a role prompt (“act as”), for instance, “act as an experienced project manager”. Some GenAI assistants offer the possibility to provide this information, also referred to as “system prompt” in a separate chat window (see e.g., the custom instructions in ChatGPT or the system prompt in the OpenAI playground). This approach is particularly beneficial when a user wants the GenAI assistant to respond in a manner consistent with a specific role or scenario. With the use of a system prompt, there is no need to repeat the same instruction in different conversations [28].

d) Apply multimodal prompting: Multimodal prompting means to provide a GenAI assistant with inputs or prompts in different modalities, such as text, images, audio and video. For example, Google's Gemini multimodal model is able to leverage the entire context of a conversation and combine written and visual information to generate its answer [29]. This technique is useful not only in tasks such as content creation, but also in complex decision-making scenarios, where input and output in various modalities provide a more complete view of the situation.

2) Understand GenAI Assistants' Capabilities and Limitations

It is important that users are aware of the actual capabilities of GenAI assistants, i.e., to know what tasks they can perform, what knowledge they use to answer queries, and how they differ from search engines. This will help users to frame their questions and expectations accordingly. Awareness of limitations and potential difficulties, such as bias in the underlying data, is also extremely important, as it is a prerequisite for users to write appropriate prompts and interpret the results provided by GenAI assistants in a responsible manner. For instance, users should be careful to write prompts that are free of pre-existing biases or assumptions about gender, race, or other sensitive factors to avoid reinforcing bias [21]. Users should think critically, verify the results with their own expertise and conduct further research in trusted sources [21].

3) Adopt an Iterative Approach

Users should be prepared to take an iterative approach when interacting with a GenAI [3]. It may be necessary to phrase the prompt in a different style or add information. Being flexible and adapting to the GenAI assistant's conversation flow will ensure a smoother conversation. Sometimes users need to be patient and try different approaches. While this can be considered a form of learning, users need to be aware of the chatbot's limitations and should consider whether not using the chatbot might be more efficient in specific situations.

4) Awareness of Privacy and Security Issues

When interacting with GenAI assistants, users should critically evaluate what information they are willing to share. They need to be aware of whether the question or contextual information provided in a prompt reveals sensitive data. This applies to both personal and business-related data. For example, users should be careful about writing project-specific details, financial information, or proprietary data. They should always formulate such information in a generalized and anonymized manner, and consider the need to disclose sensitive information.

5) Being Aware That You are Talking to an AI

Users need to be aware that they are communicating with an AI, not a human. Because of the conversational interface, users tend to think of a GenAI assistant as a human, or at least attribute some human-like characteristics to it. This can be observed in users' prompts where they use phrases like "please", "thank you", even though they know that they are talking to a bot. Anthropomorphizing the GenAI assistant poses several dangers, such as shifting accountability to the GenAI assistant or overestimating its capabilities [6].

Project managers using GenAI assistants should be aware that GenAI assistants may not be able to handle complex or nuanced situations that require human judgment and decision-making. They operate based on data and patterns, which may not always account for a specific situation or in unforeseen circumstances. Users should think critically and consider the limitations of GenAI assistants when relying on their advice or recommendations. Project managers should also understand that not only can they be misled by a GenAI assistant's response, but so can their team members.

IV. CASE STUDY ON INCORPORATING GENAI ASSISTANTS IN PROJECT MANAGEMENT

The following case study demonstrates how GenAI assistants can increase the efficiency and effectiveness of project management. It considers a digitalization project at a European manufacturing company, which aims to introduce a new software for recording the working time. This system should simplify and automate current processes. The project encompasses the vendor selection, the customization of the selected solution, the process redesign, and the training of the future users. Note that scope and objectives of the project considered in the project management case study are already defined as it would be the case in any project in an enterprise context. A conversational AI could have been used in this preliminary step, too. For instance, it could support the project initiators to better formulate the objectives or assist other tasks in project definition.

Within the case study, concrete activities in project management should be performed with the help of a GenAI assistants, namely one of the project performance domains described in the PMBOK Guide published by the Project Management Institute (PMI) [30]. A project performance domain is defined as "a group of related activities that are crucial for the effective delivery of project outcomes" [30]. The PMBOK Guide outlines eight performance domains that should be considered during the implementation of a project. These domains play a crucial role in driving the project towards its intended outcome. In the case study, the focus will be on the stakeholder performance domain.

This performance domain emphasizes that it is important to maintain alignment with the stakeholders of a project and to engage with them a positive relationship [30]. Stakeholders are persons or groups of people who affect the project and its outcomes, or who are affected by the project and its outcomes. They all bring their own ideas, values, qualifications, and prior experience [31]. The PMBOK Guide identifies several activities in the context of the stakeholder management as shown in Figure 3.

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

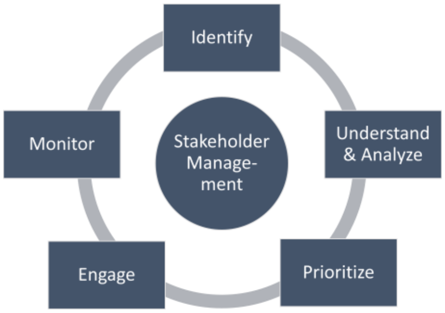


Figure 3. Activities to ensure stakeholder engagement based on [30].

A first try could be to write a single prompt that asks the GenAI assistant to perform the stakeholder management, such as “Do the stakeholder management for a digitalization project in a German manufacturing company that aims to introduce a new digital system for recording the working time.”. The result is, according to ChatGPT, a “comprehensive stakeholder management plan for your project”. It contains eight steps that can be mapped to the activities named in the PMBOK Guide. For each step, a type of deliverable is presented, such as a list of stakeholders, or a communication plan. However, a closer look at the results reveals that they are rather general. For instance, one recommendation for managing stakeholders’ expectations is to highlight the benefits of the new system, such as accurate time tracking, improved efficiency, and reduced paperwork. This is true, but too superficial to be useful in day-to-day work. This example highlights the need for more sophisticated prompts. For instance, each of the eight steps could be refined with the help of individual prompts, each focusing on specific aspects of stakeholder management. Such prompts are shown below for each stakeholder management activity.

In addition to the individual prompts, we define for this case study in ChatGPT a system prompt as shown in Table I. It frames the conversation with the GenAI assistant in the scope of this project management case study.

TABLE I. SAMPLE SYSTEM PROMPT

Intent	Let the GenAI assistant act as an experienced project manager
Type of prompt	System prompt
Required input	Description of the desired behavior
Prompt	You are an experienced project manager with certifications in various project management standards, which you are able to apply in practice. You are now responsible for the following digitalization project and you have experience with this type of project.

By specifying that the GenAI should act as an experienced project manager, we aim to increase the accuracy of its responses. For instance, the GenAI will use concepts and terminology that are specific to the field of project management.

A. Identify Stakeholders

First, it is necessary to identify all potential internal and external stakeholders of a project. While certain stakeholders can readily be identified, there might also be stakeholders that are only indirectly affected by the project [30]. They are more difficult to identify. In practice, the activity of identifying stakeholder is often performed by the project manager and some team members in a brainstorming session [32]. A GenAI assistant can support this task. To enable the GenAI assistant to generate a list of stakeholders that is specific to the project at hand, some context about the project must be provided. It is recommended to describe the project in a short but precise way. By doing this, it has to be taken care that no confidential information is passed to the AI-based system. Table II shows such a prompt.

TABLE II. SAMPLE PROMPT TO STAKEHOLDER IDENTIFICATION

Intent	Identify the stakeholders of a project
Type of prompt	Brainstorming
Required input	Context information about the project
Desired output	List of stakeholders including a brief description of how they are connected with the project.
Prompt	Consider the following project and identify the different stakeholders involved in this project: The digitalization project in a German manufacturing company aims to introduce a new digital system for recording the working time. This new system should simplify and automate the process of recording working time. This project encompasses the vendor selection, the customization of the selected solution but also the process redesign and the training of the future users. Identify the internal and external stakeholders for this project. Consider the broad scope of the project, including technical, compliance, and user adoption. For each stakeholder, provide a brief description and explain how it relates to the project.
Excerpt from the result (gpt-turbo-4-preview)	**Company Management and Executives**: These are the top-level decision-makers in the company who have approved the project. They are interested in the project's alignment with the company's strategic goals, such as increasing efficiency and reducing operational costs. **HR Department**: The human resources department is a key stakeholder, as they are directly involved in labor management, including working time recording. They will be concerned with how the new system impacts labor regulations compliance, employee contracts, and overall employee management. **Employees (Users)**: […] **IT Department**: […] **Works Council (Betriebsrat)** […] […]

Before further using the results within the project, it is necessary to carefully check the plausibility in the specific context of the project. As the GenAI assistant generates the answer based on the information seen during the training of the LLM, relationships could be created that are not relevant in the current project. However, it may also be the case that the GenAI assistant identifies stakeholders that one has not thought about oneself. So, one should carefully think about them before excluding them from the list of stakeholders. In addition, the list of stakeholders might be incomplete and might need to be extended "manually". This could be done in an AI-assisted video conference with the project team. With GenAI, key discussion points, such as specific aspects about a particular stakeholder or discussions about an additional stakeholder, could be automatically summarized. GenAI technology can also identify and attribute specific points of discussion to participants, allowing for the tracking of contributions and where participants are in agreement or disagreement.

B. Understand and Analyze Stakeholders

To be able to effectively engage and communicate with stakeholders, it is important to understand their "feelings, emotions, beliefs and values" [30]. A GenAI assistant can help to analyze the stakeholders' perspectives. In a first step, it could be asked in a rather general way to gain additional information about the stakeholders, their connection to the new system and to the project. Such a prompt could be "Provide more information about the stakeholder 'work council'.". Note that GenAI assistants remember the information given within one conversation. So, the prompts can build upon each other. Including good practices from project management in the prompt can improve the output as shown in the prompt in Table III.

TABLE III. SAMPLE STAKEHOLDER ANALYSIS PROMPT

Intent	Understand the stakeholders of a project
Type of prompt	Generate insights
Required input	The stakeholder to be analyzed. If available, context information about the stakeholder or past experiences with this stakeholder should be added.
Desired output	Information about each of the key aspects required in the prompt. Note that without providing the key aspects in the prompt, the result would be more general.
Prompt	Discuss for the stakeholder "work council" its power, impact, attitude, beliefs, expectations, degree of influence, proximity to the project and its interest in the project.
Excerpt from the result (gpt-turbo-4-preview)	The **Works Council (Betriebsrat)** plays a significant role in German companies, acting as an intermediary between the employees and management. […] Let's break down these attributes: ### Power […]

This example shows how certain aspects that the PMBOK Guide recommends to consider can be added as desired outputs to the prompt. In the stakeholder analysis, the PMBOK Guide specifically advises reviewing the stakeholders' power, influence, attitudes, beliefs, expectations, degree of influence, proximity to the project, and interest in the project [30]. The project team should review the results of the prompt as it might be necessary to refine them based on project specific information.

To deepen the understanding of the stakeholders' perspectives, fictional characters, so-called personas, who represent the respective stakeholder can be used [32]. Persona analysis is also known in marketing and innovation management. Bilgram and Laarmann demonstrate the generation of a marketing persona in their ChatGPT use case from the automotive industry [18]. In our case study, the GenAI assistant could be asked to describe typical personas that represent the stakeholder "employees", including their names, their professions, and the impacts that the new system would have on them. Then, the virtual assistant could be asked in a follow-up prompt what a certain persona would tell us about the project providing insights into the individual perspectives, concerns, and expectations. In the results we obtained, it was necessary to ask explicitly for critical perspectives. The answers to our first prompts rather sounded coming from a marketing brochure of time tracking systems and every persona was said to be happy with the system. Table IV shows a prompt that provides us with a more critical perspective.

TABLE IV. SAMPLE PROMPT WITH PERSONAS

Intent	Explore different perspectives
Type of prompt	Generate insights
Required input	Additional information about the personas, such as their current way of working and potential concerns
Desired output	Differentiated opinion from the persona's perspective
Prompt	What would persona <A> say about the time tracking system when she was formerly used to start and stop working whenever she wished, only being judged by her results, not her time spent on them?
Excerpt from the result (gpt-turbo-4-preview)	[…] Here is a possible reflection of their perspective: […] "My initial concern is that it shifts the focus from the quality of work produced to the sheer hours logged, which feels like a step backwards in terms of workplace modernization and individual autonomy. I've always valued the trust placed in me to manage my time effectively, working when I feel most productive—whether that's early in the morning or later in the evening—without being tied to a conventional 9-to-5 schedule. […]"

Alternatively, we could have prompted the GenAI assistant with a clear instruction about the persona's feeling towards the project such as "Create a persona for the stakeholder group employee who is quite critical with regard to this new project." Based on these results, a follow-up prompt could be to generate ideas on how to address potential concerns, such as "What incentives could be used to raise acceptance?".

C. Prioritize Stakeholders

Due to limited resources, projects normally need to focus on the most important stakeholders [30]. Therefore, it is common to evaluate all stakeholders with respect to their attitude to the project, i.e., whether they have a positive or negative interest in the project and its outcomes, and whether they have a high or low influence on the project's progression, i.e., their power [30], [31]. Besides of interest and influence, other criteria can be used, such as whether the stakeholders have already determined their opinion (and we will find it difficult to change it), or whether their opinion is still open and can be influenced. Discussing different criteria for a stakeholder matrix is beneficial because it allows for a more nuanced and comprehensive analysis of stakeholders and their attitudes towards the project. On this basis, appropriate measures can be taken to effectively involve these stakeholders in the project. Prioritizing the project's stakeholders is also important for managing conflicting interests.

Having analyzed the stakeholders according to such criteria, a so-called stakeholder matrix or stakeholder map can serve as a tool to provide an intuitive visual overview [33]. The x- and y-axes of the stakeholder matrix are labelled with the criteria, e.g., interest and influence, and the stakeholders are placed into one of the four quadrants. Other common visualizations can also include four quadrants and positive/negative criteria values. Figure 4 shows an example of a stakeholder matrix.

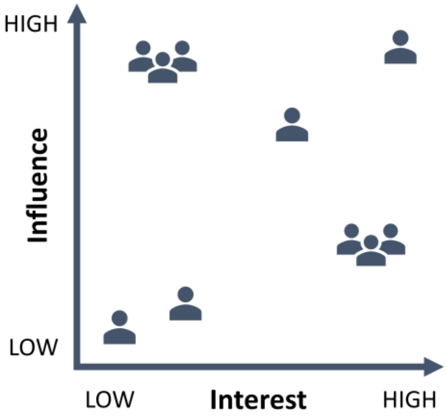


Figure 4. Sample stakeholder matrix.

In the following, we show a step-by-step approach that aims to create such a graphical representation with the help of GenAI. Note that you could also use a simple prompt such as "Classify the stakeholders identified for this digitalization project in the different quadrants of a stakeholder matrix" to get first results and to create the stakeholder matrix manually. This typically leads to a classification that uses the most common factors, i.e., interest and power.

The following tables demonstrate how such a stakeholder matrix can be created in several steps. Table V uses the GenAI assistant to get some advice, which criteria might be appropriate for the stakeholder matrix in the project at hand.

TABLE V. SAMPLE PROMPT SELECTING PRIORITIZATION CRITERIA

Intent	Select axes for the stakeholder matrix
Type of prompt	Get advice
Required input	Clear instructions regarding the required ouput, i.e., suggestions for the axes of a stakeholder matrix. Instruction that the choice of the axes should refer to the project in order to avoid a general statement.
Desired output	Two combinations, where for each combination the two axes are explained and the combination is motivated
Prompt	We have identified the stakeholders for this project. Now, we want to create a stakeholder matrix. Identify at least two appropriate combinations of x- and y-axes for the stakeholder matrix and explain for each combination why itis well suited to the project at hand.
Excerpt from the result (gpt-turbo-4-preview)	[…] ### Combination 1: Influence (Y-Axis) vs. Interest (X-Axis) […] **Why This Combination?** This matrix helps prioritize stakeholders based on their power to affect the project and their level of concern about the project's outcome. It is particularly suited for this project because: 1. It identifies stakeholders who need close management (high interest, high influence) versus those who need merely to be kept informed. 2. It aids in developing a nuanced communication plan that balances the needs and concerns of all stakeholders, ensuring that those who could potentially derail the project are managed proactively. ### Combination 2: Support (Y-Axis) vs. Impact (X-Axis) […] **Why This Combination?** This orientation offers insight into managing stakeholder relationships and resistance to change, crucial for a project that involves significant process alteration and digital adoption: 1. It highlights stakeholders who could be leveraged as change agents versus those who might require targeted change management interventions to mitigate resistance. 2. It is instrumental in the planning phase to ensure that those most affected by the project are adequately supported, and their feedback is incorporated into the project design, thus enhancing user adoption and project success.

In the next step shown in Table VI, we define the x- and y-axes and ask the GenAI assistant to classify each stakeholder according to these criteria and explain the reasoning behind the classification. The resulting prioritization can be used as a basis when planning measures for engaging and communicating with stakeholders. For instance, it is important to address any concerns that stakeholders with high power and high interest have. Note that the prompt provides quite detailed instructions about the matrix, the scale of the axes and the quadrants, in which the

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

stakeholders should be classified. The prompt asked for concrete values as we wanted the GenAI assistant to provide its response in a structured format that can be easily compared, analyzed, and visualized.

TABLE VI. SAMPLE PROMPT TO CLASSIFY STAKEHOLDERS

Intent	Create a stakeholder matrix
Type of prompt	Create project documentation
Required input	• List of stakeholders and their concerns • Dimensions of the stakeholder matrix • Scale of the axes
Desired output	Classification of the stakeholders in the four quadrants of the predefined stakeholder matrix.
Prompt	I want to classify the stakeholders identified for this digitalization project in the four quadrants of a stakeholder matrix. The first dimension (x-axis) should be the interest and the second dimension the influence (y-axis). The scale is for each axis from 0 to 10. Provide for each stakeholder the values for both dimension and explain your reasoning. Group the stakeholders per quadrant, i.e., High Influence, High Interest; High Influence, Low Interest; Low Influence, High Interest; Low Influence, Low Interest. The response format is as follows for each stakeholder: # Stakeholder ## Interest: <value> < provide a detailed reasoning here > ## Influence: <value> < provide a detailed reasoning here >
Excerpt from the result (gpt-turbo-4-preview)	[…] ### Low Influence, High Interest **6. Employees (Users)** - **Interest**: 9 The new system directly impacts their daily work routines and job satisfaction. - **Influence**: 4 Despite their direct interaction with the new system, their ability to influence project decisions or outcomes is limited. […]

Interestingly, the GenAI assistant classified stakeholders with a value of 6 (on a scale from 0 to 10) in the dimension influence into the quadrant with low influence. For instance, the AI GenAI assistant assigned to the stakeholder work council an interest of 8 and an influence of 6, and positioned it in the quadrant Low Influence / High Interest. Asking the GenAI assistant for the reasoning behind this classification, it explained that the classification reflects a comparative assessment rather than an absolute measure. Stakeholders with scores in the upper quartile (e.g., 8, 9, 10) were classified as high interest/influence. In contrast, stakeholders with scores in the lower quartile (e.g., 1, 2, 3) were classified as low. Those in the middle were classified depending on the overall distribution of scores. Without further explanation, this approach can be very disconcerting for project participants working with the stakeholder matrix.

The step presented in Table VI could easily be repeated with a different combination of x- and y-axes in the stakeholder matrix by adjusting the prompt. To continue working with this information, i.e., with the values that are determined for the different axes, it might be useful to ask the GenAI to sum it up in a table or to provide the information as comma-separated values (CSV).

The goal of the next step was to create a visual representation of the stakeholder matrix based on the detailed information generated in the previous step, placing each stakeholder name or group in one of the four quadrants. Within the case study, different options were evaluated.

The first option was the creation of a visual representation with the help of an GenAI assistant that is combined with image generation tools (such as ChatGPT 4.0 Plus with Dall-E or the Microsoft Copilot, which includes the Image Creator from Designer, formerly Bing Image Creator). The results, however, were far below our expectations. Figure 5 shows one of the generated images. For instance, the labels are not correct but contain many spelling errors. The stakeholders, which were specified in the prompt with exact coordinates were not included in the stakeholder matrix but only generic representations of persons.

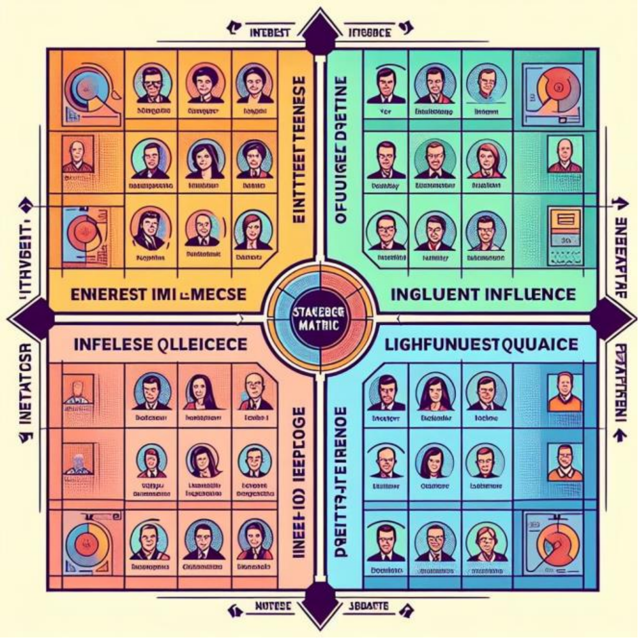


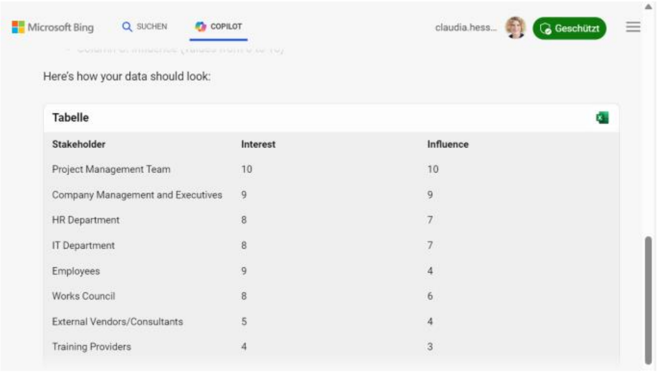
Figure 5. Stakeholder matrix generated by Bing Image Creator (created 12/02/2024).

The second option that was evaluated in the case study was to ask the GenAI assistant to generate the diagram based on given data in Microsoft Excel. We selected Microsoft Copilot as tool because, in contrast to ChatGPT, it is able to generate an Excel file. We tried different prompts, but Copilot was not able to create the diagram directly in the Excel file but only the table that can be used as a basis for creating a diagram. Table VII shows the approach.

TABLE VII. SAMPLE PROMPT TO GENERATE A CHART IN MICROSOFT EXCEL

Intent	**Create diagram for the stakeholder matrix in Excel**
Type of prompt	Create project documentation
Required input	• The tool to be used, i.e., Excel • Data to be visualized
Desired output	Excel file with a chart
Prompt	Create a chart in Excel that is appropriate to represent the following information as a stakeholder matrix. The x-axis represents the interest and the y-axis represents the influence. The scale goes from 0 to 10. We have these data points: Stakeholder (Interest,Influence) Project Management Team (10,10) Company Management and Executives (9,9) HR Department (8,7) IT Department (8,7) Employees (9,4) Works Council (8,6) External Vendors/Consultants (5,4) Training Providers (4,3) Regulatory Bodies (3,8) Unions (6,5)
Excerpt from the result (Microsoft Copilot)	Certainly! Let’s create a **Stakeholder Matrix Chart** in Excel based on the provided data points.. […] 1. Open Microsoft Excel. 2. Enter the data points in three columns: Column A: Stakeholder names […] Here’s how your data should look: [Embedded Excel] 3. Highlight the data (including headers) in columns A, B, and C. 4. Go to the “Insert” tab and choose “Scatter” from the “Charts” group. 5. Select “Scatter with Straight Lines” (the first option). Your scatter plot will now be displayed. You can customize the chart by […]

At a first glance, the answer seems to be useful. The GenAI assistant creates a downloadable Excel file as shown in Figure 6. Copilot briefly describes the steps needed to create a what it calls “stakeholder matrix chart”. It seems to efficiently guide us through the process, even though it is unable to create the chart itself. However, a closer look at the steps reveals that the GenAI assistant recommends using a scatterplot with straight lines. This is neither a common nor a useful representation of the stakeholder matrix. A scatter chart without lines or a bubble chart would have been a better choice. Moreover, when creating the scatter chart, one should not select columns A, B and C, but only B and C in order to display interest and influence as x- and y-axes. The values in column A must be selected in an additional step as labels for the bubbles.



Here’s how your data should look:

Tabelle

Stakeholder	Interest	Influence
Project Management Team	10	10
Company Management and Executives	9	9
HR Department	8	7
IT Department	8	7
Employees	9	4
Works Council	8	6
External Vendors/Consultants	5	4
Training Providers	4	3

Figure 6. Excel file by Bing Image Creator (created 12/02/2024).

A further option could be to ask the GenAI assistant to generate a MATLAB script to create a chart visualizing the stakeholders in the described stakeholder matrix format. When discussing such alternatives, it is important to keep in mind that in order to serve as a communication tool, the visual representations need to be visually appealing. Therefore, we recommend that you make a manual effort to customize them to suit your needs.

The list of stakeholders and the stakeholder matrix are important artefacts created within project management and which are used during project meetings, presentations, or when sharing project updates. All project documentation regarding the stakeholders should be reviewed throughout the course of the project, and, if necessary, be updated because on the one hand, stakeholders might change their opinions and, on the other hand, influencing factors might change [30]. In addition, misevaluations can be corrected. A further idea is to use the GenAI assistant to prepare the information for a so-called onion diagram, which shows the relationship between the stakeholders and the solution that is developed in the project [32]. Like the layers of an onion, the stakeholders in the center of the graphic are those who are directly involved in developing the new solution. Stakeholders on the outer layers are less affected by the solution.

Summarizing the experiences in this step, we can say that while GenAI assistants help us analyze and prioritize stakeholders, they fail to provide us with a graphical representation of the stakeholder matrix for the time being. However, this would be very important as it is an effective tool to communicate complex information about stakeholders in a clear and concise manner.

D. *Engage Stakeholders*

Successful stakeholder engagement requires different communication approaches [30]. Depending on the audience and the type of information, a verbal or written communication should be preferred. Moreover, the communication might be formal or informal. An example for a formal written communication is a progress report, while a social media post or an instant message is an example for a written informal communication.

A GenAI assistant can support project managers in preparing an appropriate communication artefact. This might include several steps, in which the content of the communication is elaborated, the type of communication is determined, and the actual text is written. By doing this, users need to take care that the communication does not remain superficial and that it does not appear to be artificially generated. The time saved by utilizing a GenAI assistant can be used to create customized communications for each stakeholder, which can also be supported by the GenAI assistant.

In the following, we show different steps to create content targeted at the communication with a certain stakeholder. First, we should think about the communication strategy. Table VIII shows the first prompt, which provides the project management with ideas and advice regarding the communication with a certain stakeholder. Note that the results generated by the GenAI assistant are very detailed due to the precise instructions that were given in the prompt, i.e., the four questions.

TABLE VIII. SAMPLE PROMPT ABOUT THE COMMUNIATION STRATEGY

Intent	Plan how to communicate with a specific stakeholder group
Type of prompt	Brainstoring
Required input	Additional information about the stakeholder and the topic to be communicated
Desired output	Ideas as basis for planning communication measures
Prompt	We have identified the stakeholders in this project. An important stakeholder are the employees. They are the end-users of the new system. Their cooperation is vital for the successful adoption of the system. Their concerns would typically revolve around ease of use, accessibility, privacy issues, and how the system might affect their daily routines and job responsibilities. We want to engage the employees by a targeted communication. Generate ideas based on the following questions. #How should the rationale behind the new system be communicated to the employees? # How should the communication be realized? # Who should communicate in which way? # How often?
Excerpt from the result (gpt-turbo-4-preview)	In implementing a successful communication strategy for introducing the new digital time recording system to the employees of the German manufacturing company, consider the following comprehensive approach: ### **1. Communicating the Rationale Behind the New System:** - **Initial Announcement:** Start with an all-hands meeting (either in-person or virtual, considering the size and location of your workforce) to announce the introduction of the new system. The announcement should convey the benefits clearly, such as increased accuracy, transparency, reduced administrative burden, and how it aligns with broader organizational goals like efficiency and digital transformation. - **Q&A Sessions:** Follow up with Q&A sessions where employees can express their concerns and get immediate responses. This will help in addressing misinformation and easing anxieties right from the start. ### **2. How the Communication Should Be Realized:** - **Multichannel Approach:** Utilize various communication channels to ensure the message reaches everyone effectively. This includes company-wide emails, intranet posts, newsletters, and physical posters in common areas. Tailoring the message for each medium increases engagement and comprehension. - **Transparent and Continuous Updates:** Provide ongoing updates about the project's progress, what stages are completed, and what's coming next. This transparency builds trust and keeps the project on everyone's radar. ### **3. Who Should Communicate and in Which Way:** - **Leadership Involvement:** The message should first come from the top - the CEO or the Plant Manager can introduce the initiative, underscoring its importance to the company's future. This shows the company's commitment from the highest level. - **Project Team and Department Heads:** Regular, detailed communication should come from the project team, facilitated by department heads. They are closer to the employees' daily operations and can provide a more relatable context. Department heads can also gather feedback and concerns to be addressed by the project team. - **Dedicated Champions:** Identify and train 'system champions' from among the employees who can assist their peers, offer additional training, and act as a feedback loop to the project team. ### **4. Communication Frequency and Feedback Loops:** […]

Based on this information, concrete communication measures can be planned. It is of course possible to go into more detail on certain aspects with further prompts. In the case study, we follow the recommendations by the GenAI assistant. An all-hands meeting should be organized, in which the new system is announced. The GenAI assistant can also help with this. Table IX shows a prompt that prepares the text for invitation to be sent to the employees. The resulting text could be used as base. The GenAI assistant provided some placeholders to be filled with the actual date, location etc.

TABLE IX. SAMPLE PROMPT TO GENERATE COMMUNICATION MATERIALS

Intent	Engage stakeholder by a targeted communication
Type of prompt	Create communication materials
Required input	Additional information about the stakeholder and the topic to be communicated
Desired output	Text
Prompt	The employees should be invited to an all-hands meeting to be held in person at our office. Write the invitation email for this meeting on behalf of the CEO.
Excerpt from the result (gpt-turbo-4-preview)	Subject: Invitation to All-Hands Meeting: Introducing Our Next Big Step Forward Dear Team, I hope this message finds you well. As we continue on our journey of growth and improvement, it’s crucial that we stay connected, informed, and aligned with our company’s mission and the innovative strategies that drive us forward. In line with our commitment to innovation and operational excellence, we are excited to embark on a new project that promises to enhance our efficiency and streamline our processes - the introduction of a new digital system for recording working time. This system represents a significant step towards modernizing our operations, ensuring accuracy in time management, and providing everyone with a more intuitive and user-friendly way to record their working hours. To share more about this pivotal project, discuss its benefits, and address any questions you may have, I warmly invite you to an All-Hands Meeting: **Date:** [Insert Date] **Time:** [Insert Time] **Venue:** [Insert Venue Name/Location at the Office] **Agenda:** 1. Introduction to the New Digital Time Recording System 2. Rationale and Benefits 3. Implementation Plan Overview 4. Training and Support 5. Q&A Session This meeting will also be an opportunity for us to come together as a team, share our thoughts, and ensure that everyone feels supported and heard as we move forward with this change. Your input and engagement are invaluable to us, and I encourage each one of you to attend. Please RSVP by [Insert RSVP Deadline] to ensure we have adequate arrangements for everyone. Should you have any preliminary questions or topics you’d like to see covered, feel free to reply directly to this email or reach out to [Insert Contact Person’s Name and Email]. Your dedication and hard work is what drives our success. Together, I am confident we’ll make this transition smoothly and continue to achieve great things. Looking forward to seeing all of you there and embarking on this exciting journey together. Best regards, [CEO’s Name] CEO [Company Name]

Note that the better the AI chatbot knows someone’s writing style, the better it could adapt the wording and the tone. So, it might be a good idea to provide some sample texts. However, these should only be openly accessible texts if you are using a public GenAI assistant to avoid inadvertently revealing sensitive or personal information. An alternative that could use multimodal models would be to generate ideas and illustrations for physical posters in common areas, as suggested by the GenAI assistant in the sample prompt in Table VIII about the communication strategy.

A further application scenario in stakeholder engagement is the training and simulation of difficult situations. For example, a GenAI assistant can help the project manager to prepare for a project presentation with stakeholders who express many concerns about the project. The project manager can role-play a discussion about the project, try out different lines of argumentation and train to respond to difficult questions. Moreover, role-playing with a GenAI assistant could include scenarios that specifically aim to better understand certain perspectives and to foster empathy. Using AI-driven simulations for training will evolve beyond text-based interactions to include video-based scenarios, making the training experience even more realistic.

To continue with the example from our case study, the project manager could use the GenAI assistant to prepare for the Q&A session at the all-hands meeting. Talking to the GenAI assistant provides a safe and controlled environment for training. Table X demonstrates such a prompt. We explicitly ask the GenAI assistant to act as a stakeholder who has a highly critical and skeptical view of the project and therefore asks critical questions and follow-up questions. The GenAI assistant could also be prompted to act as a stakeholder who becomes very emotional. This setup can simulate scenarios where emotions run high, such as discussions about project goals, resource allocations, or changes that significantly impact stakeholder interests. As we can see in this short dialog, the GenAI assistant did not accept the first, very superficial answer given by the project manager and asked for more details. The project manager responds by referring to an ISO standard that is not about IT security, but about project management. The GenAI assistant, trained to be a friendly assistant, continues in a very professional tone and requests more details.

TABLE X. SAMPLE PROMPT FOR TRAINING

Intent	Train the conversation with a difficult stakeholder
Type of prompt	Training and simulation
Required input	Information about the situation
Desired output	Flipped interaction with questions asked by the GenAI assistant and answers given by the user
Prompt	As the project manager, I want to prepare myself for this meeting. Act as an employee asking critical questions about the project. I want you to have a dialogue with me about your concerns. Start with asking a certain question. Allow me to reply before moving on. If my answer to your question is not convincing, then ask critical follow-up questions. Do not write explanations, and reply only as the employee. Our conversation starts with me. Project Manager: "We're going to start our Q&A session now. What questions do you have about the project?"
Excerpt from the result (gpt-turbo-4-preview)	Employee: "Thank you for the presentation. My first question is, how will this new digital time recording system ensure the privacy and security of our personal data?" Project Manager: "You can be sure that we take all necessary measures to ensure that your data is secure." Employee: "Could you specify what kind of security measures are being implemented? Are there any certifications or standards the chosen system complies with?" Project Manager: "The system will be ISO 21500 compliant. By following this standard, we ensure a wide range of security measures during implementation and when the system is in production." Employee: "I understand the ISO compliance part for project management, but what about data protection specifically? Is the system GDPR compliant, and how are we addressing the right to privacy under this regulation?"

Having completed a conversation with a certain line of argumentation, the project manager can discuss it on the meta-level with the GenAI assistant. The GenAI assistant could provide insights into the effectiveness of different communication strategies, helping project managers identify areas for improvement. Adapting the example from [34], a prompt could be: "Pause the dialogue. For each of my answers, show me how I can improve it in order to make it more convincing."

E. Monitor Stakeholders

During a project, stakeholders might change their opinions, new stakeholders might become relevant while the importance of others may decline. Therefore, it is important to monitor the stakeholders and their satisfaction with the project [30]. GenAI assistants can support by analyzing the sentiments that stakeholders express in their communication. E-mails or any other communication in internal groups or communication platforms about the project could be analyzed by a GenAI assistant. The aim is to gain an understanding of the attitudes, opinions, and emotions expressed by the communicators. Table XI shows the first step in such an analysis, which is to assess the sentiments expressed in communications, in this example, in social media posts.

TABLE XI. SAMPLE PROMPT TO SENTIMENT CLASSIFICATION

Intent	Classify the sentiment in social media posts
Type of prompt	Get insights
Required input	Communication to be analyzed, here social media posts. Note that we use artificially created social media posts in this example to demonstrate the approach.
Desired output	Categorization of each social media post as positive, negative, or neutral based on the content's emotional tone
Prompt	Classify the sentiment of each of the following social media posts as positive, negative, or neutral. [social media posts]
Excerpt from the result (gpt-turbo-4-preview)	### Post 1: From a Tech-Savvy Young Employee - **Sentiment**: Positive - This post radiates enthusiasm and optimism about moving to a digital system for recording working hours. The mention of "exciting times," the anticipation of streamlined processes, and hope for user-friendliness indicate a positive outlook. ### Post 2: From a Long-Term Employee Resistant to Change - **Sentiment**: Negative - The post reflects resistance to change, skepticism about the efficacy of training, and a preference for the status quo. Phrases like "if it ain't broke, why fix it?" and expressing doubts highlight a negative sentiment. […]

Based on these results, the GenAI assistant could be asked about the characteristics of the dataset, for example, with the prompt "Aggregate the sentiment scores to quantify overall sentiment for the entire dataset. Determine the proportion of positive, negative, and neutral sentiments." This distribution is an indication of the tendency of people's opinions about the project. To identify the specific themes or issues driving stakeholder sentiment within the highly positive and negative posts, one could analyze with the help of the GenAI assistant the context and content of these posts more closely.

Combining this approach with the capacities of search engines, it would be possible to ask about stakeholder opinions and reactions to specific actions in other, similar projects. It is important to ensure that no sensitive data is entered and that confidentiality regulations are observed.

V. POTENTIALS AND RISKS OF USING GENAI ASSISTANTS IN PROJECT MANAGEMENT

The pattern of AI-assisted project management activities, as well as the examples within the case study have demonstrated that GenAI assistants can indeed be a useful digital tool in project management. However, there are several issues that lead us to conclude that they are "just" a tool and not yet a digital assistant that is a full member of the project team. First, project managers need to trust that the GenAI assistant has a solid knowledge of project management methods and good practices, for instance, when asking for guidance. To build up this trust, it would be helpful if the GenAI assistant added a reference to established project management standards, such as published by PMI, ISO or PRINCE2 to the answer. As long as this is not the case, it is recommended to provide this guidance to the GenAI assistant in the prompt, such as shown in the prompt in Table II, which explicitly mentions the different aspects to be considered in the stakeholder analysis.

Second, the project manager must be able to review any generated content. For instance, if the GenAI assistant extracts a project status from various documents, traceability is required. In other words, it must be clear what information was used to set the status. To a certain extent, this can be achieved by explicitly asking the GenAI assistant to provide its reasoning in addition to the answer. This can be achieved, for instance, by adding a specific response format that guides the GenAI assistant such as shown in Table XII.

TABLE XII. SAMPLE PROMPT FOR PROVIDING REASONING

Intent	**Obtain the GenAI assistant's reasoning in addition to the answer**
Type of prompt	Transparency
Required input	Information about the situation
Prompt	[Prompt] Response format: # Reasoning < provide a detailed reasoning here > # Response < Provide your response here >
Result	Answer splitted into two parts, the reasoning behind the response and the response itself

Third, as the example prompts have shown, the more specific details about the project, its environment, and stakeholders are provided, the better the GenAI assistant's responses will be. However, this is likely to conflict with data protection and privacy. Users should be extremely careful about the disclosure of any sensitive information. Not only is there the problem of the GenAI assistant adding this information to its knowledge base and using it to generate answers, but it also provides the GenAI assistant with information to infer ("to guess") further sensitive information. A solution might be to further develop GenAI assistant as business-internal solutions, which are separated from the outside world. In this case, the underlying LLMs could be trained with company-internal data, e.g., on past projects. This would provide the GenAI assistant with company-specific knowledge and push it a step further to a digital assistant.

VI. CONCLUSION AND FUTURE WORK

The paper explored the possibilities of using an GenAI assistant, such as ChatGPT, to support project managers in their daily work. Patterns of AI-assisted project management tasks were identified, showing the wide range of possible collaborations between project managers and GenAI assistants. Examples include using the GenAI assistant to generate ideas and insights, or to act as a sparring partner. The skills required for successful collaboration were also presented. Special emphasis was placed on prompt engineering. The ability to write concise prompts is crucial to achieving results that are useful and not superficial. A case study demonstrated how a GenAI assistant could be used. Concrete prompts have shown how a GenAI assistant can support stakeholder management activities in a variety of ways. Examples include prompts that make certain tasks easier, such as writing the text for an invitation to a meeting, or asking the GenAI assistant for advice on creating a stakeholder matrix. We also demonstrated how the GenAI assistant can support creative activities and act as a sparring partner in training and simulations. In order to assess whether the 24/7 GenAI assistant is indeed a reliable assistant, potentials, risks and solutions were discussed for all application scenarios.

It would be interesting to follow project managers as they begin to use a GenAI assistant, and see how it evolves. In such a study, the project managers' learning curve associated with the use of GenAI assistants in project management could be analyzed. This could also include reflection activities on the interaction between human and virtual assistants in the project context. Learning activities, for instance with regard to prompt engineering could be developed to facilitate the further use of such digital assistants. A long-term study would also be appropriate to better understand the impact of using GenAI assistants on productivity, project outcomes, team dynamics. and especially on the role of project managers.

Another line of research could be the seamless integration of GenAI assistants with existing project management software and tools. This would change the way project managers interact with project management software. AI-driven insights and recommendations could be embedded directly into project management workflows.

REFERENCES

[1] C. Hess and S. Kunz, "Could ChatGPT be Your New Project Management Assistant? Using AI Chatbots to Support Project Management Tasks" Advances on Societal Digital Transformation (DIGITAL 2023) IARIA, Sep. 2023, pp. 1–7.

[2] T. Wu, "Digital project management: rapid changes define new working environments," Journal of Business Strategy, vol. 43(5), pp. 323–331, 2022, doi: 10.1108/JBS-03-2021-0047.

[3] T. Teubner, C. M. Flath, C. Weinhardt, W. van der Aalst, and O. Hinz, „Welcome to the Era of ChatGPT et al.," Business & Information Systems Engineering, vol. 65(2), pp. 95–101, 2023, doi: 10.1007/s12599-023-00795-x.

[4] A. Nieto-Rodriguez and R. Viana Vargas, "How AI Will Transform Project Management," Harvard Business Review. https://hbr.org/2023/02/how-ai-will-transform-project-management [retrieved: May 2024].

[5] Institute of Project Management, “What Can ChatGPT Do for Project Management?, “ 2023, https://www.projectmanagement.ie/blog/what-can-chatgpt-do-for-project-management/ [retrieved: May 2024].

[6] L. Weidinger et al., “Taxonomy of Risks posed by Language Models,” Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 2022, pp. 214–229, doi: 10.1145/3531146.3533088.

[7] E. M. Bender, T. Gebru, A. McMillan-Major, and S. Shmitchell, “On the Dangers of Stochastic Parrots,” Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 2021, pp. 610–62, doi: 10.1145/3442188.3445922.

[8] A. Kulkarni, A. Shivananda, A. Kulkarni, and D. Gudivada, Applied Generative AI for Beginners: Practical Knowledge on Diffusion Models, ChatGPT, and Other LLMs. Berkeley, CA: Apress, 2023. doi: 10.1007/978-1-4842-9994-4.

[9] S. Feuerriegel, J. Hartmann, C. Janiesch, and P. Zschech, “Generative AI,” Bus Inf Syst Eng, vol. 66(1), pp. 111–126, Feb. 2024, doi: 10.1007/s12599-023-00834-7.

[10] A. Chernyavskiy, D. Ilvovsky, and P. Nakov, “Transformers: ‘The End of History’ for Natural Language Processing?,” Machine Learning and Knowledge Discovery in Databases, ECML PKDD 2021, Lecture Notes in Computer Science, vol. 12977, 2021, doi: 10.1007/978-3-030-86523-8_41.

[11] B. Agüera y Arcas, “Do Large Language Models Understand Us?,” Daedalus, vol. 151(2), pp. 183–197, 2022, doi: 10.1162/daed_a_01909.

[12] S. Mitrović, D. Andreoletti, and O. Ayoub, “ChatGPT or Human? Detect and Explain. Explaining Decisions of Machine Learning Model for Detecting Short ChatGPT-generated Text,” 2023, arXiv preprint available at https://arxiv.org/pdf/2301.13852 [retrieved: May 2024].

[13] J. White et al., “A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT,” 2023, arXiv preprint available at http://arxiv.org/pdf/2302.11382v1.

[14] L. S. Lo, “The CLEAR path: A framework for enhancing information literacy through prompt engineering,” The Journal of Academic Librarianship, vol. 49(4), 2023, https://doi.org/10.1016/j.acalib.2023.102720.

[15] A. Barcaui and A. Monat, “Who is better in project planning? Generative artificial intelligence or project managers?,” Project Leadership and Society, vol. 4, p. 100101, Dec. 2023, doi: 10.1016/j.plas.2023.100101.

[16] Q. Abbas, W. Younus, S. Malik, and M. H. Hassan, “Incorporating ChatGPT in Software Project Management,” International European Extended Enablement in Science, Engineering & Management (IEEESEM), vol. 11(9), 2023.

[17] S. Frieder *et al.*, “Mathematical Capabilities of ChatGPT,” Jul. 2023, arXiv preprint available at doi: 10.48550/arXiv.2301.13867.

[18] V. Bilgram and F. Laarmann, “Accelerating Innovation With Generative AI: AI-Augmented Digital Prototyping and Innovation Methods,” IEEE Engineering Management Review, vol. 51(2), pp. 18–25, 2023, doi: 10.1109/EMR.2023.3272799.

[19] T. Taulli, Generative AI: How ChatGPT and Other AI Tools Will Revolutionize Business. Berkeley, CA: Apress, 2023. doi: 10.1007/978-1-4842-9367-6.

[20] R. Müller, G. Locatelli, V. Holzmann, M. Nilsson, and T. Sagay, “Artificial Intelligence and Project Management: Empirical Overview, State of the Art, and Guidelines for Future Research,” Project Management Journal, p. 87569728231225198, Jan. 2024, doi: 10.1177/87569728231225198.

[21] L. Giray, “Prompt Engineering with ChatGPT: A Guide for Academic Writers,” Annals of Biomedical Engineering. Advance online publication, 2023, doi: 10.1007/s10439-023-03272-4.

[22] L. Velush, “Recapping Our Meetings at Microsoft with AI and Microsoft Teams Premium,” Inside Track Blog, https://www.microsoft.com/insidetrack/blog/how-were-recapping-our-meetings-with-ai-and-microsoft-teams-premium-at-microsoft/ [retrieved: May 2024]

[23] Z. Wang, Q. Xie, Z. Ding, Y. Feng, and R. Xia, “Is ChatGPT a Good Sentiment Analyzer? A Preliminary Study,” 2023, arXiv preprint available at http://arxiv.org/pdf/2304.04339v1.

[24] T. Taulli, ChatGPT and Bard for Business Automation: Achieving AI-Driven Growth. Berkeley, CA: Apress, 2023. doi: 10.1007/978-1-4842-9852-7.

[25] Y. Song et al., “A Communication Theory Perspective on Prompting Engineering Methods for Large Language Models,” 2023, arXiv preprint available at doi: 10.48550/arXiv.2310.18358.

[26] T. Wu, M. Terry, and C. J. Cai, “AI Chains: Transparent and Controllable Human-AI Interaction by Chaining Large Language Model Prompts,” CHI Conference on Human Factors in Computing Systems, New Orleans LA USA: ACM, Apr. 2022, pp. 1–22. doi: 10.1145/3491102.3517582.

[27] G. Marvin, N. Hellen, D. Jjingo, and J. Nakatumba-Nabende, “Prompt Engineering in Large Language Models,” Data Intelligence and Cognitive Informatics (ICDICI 2023) Algorithms for Intelligent Systems. Springer, Singapore, doi: 10.1007/978-981-99-7962-2_30.

[28] H. Tamenaoul, M. E. Hamlaoui, and M. Nassar, “Prompt Engineering: User Prompt Meta Model for GPT Based Models,” Artificial Intelligence, Data Science and Applications, vol. 838, 2024, pp. 428–433, doi: 10.1007/978-3-031-48573-2_61.

[29] A. Chen, “How it’s Made: Interacting with Gemini through multimodal prompting,” Google for Developers. https://developers.googleblog.com/2023/12/how-its-made-gemini-multimodal-prompting.html [retrieved: February, 2024]

[30] Project Management Institute, “A guide to the Project Management Body of Knowledge (PMBOK® Guide; 7th ed.),” 2021.

[31] R. J. Martinelli and D.Z. Milosevic, Project Management ToolBox: Tools and Techniques for the Practicing Project Manager, 2nd ed., Wiley, 2016.

[32] Project Management Institute, PMI Guide to Business Analysis, 2017.

[33] O. Schibi, “Managing stakeholder expectations for project success: A knowledge integration framework and value focused approach,” 2016, J. Ross Publishing.

[34] C. Hess and S. Kunz, “‘ChatGPT, Can you Motivate my Learners?’ - Co-Creation of Inspiring STEM Lessons with AI Chatbots,” ICERI2023 Proceedings - 16th annual International Conference of Education, Research and Innovation, IATED, 2023, pp. 6103–6110, doi: 10.21125/iceri.2023.1522.

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Verifiable Labels for Digital Services: A New Approach to Phishing Detection

Maël Gassmann
School of Engineering and Computer Science
Bern University of Applied Sciences
Biel/Bienne, Switzerland
email: mael.gassmann@bfh.ch

Annett Laube
School of Engineering and Computer Science
Bern University of Applied Sciences
Biel/Bienne, Switzerland
email: annett.laube@bfh.ch

***Abstract*—Users often feel unsafe and unsecure when using digital services. For normal users lacking a technical background, it is difficult to recognize a website's legitimacy. This makes them vulnerable to cyberthreats such as phishing attacks. In order to solve this issue, many organizations use corporate designs or logos to guide users through their websites. However, these files can be easily copied. More technical means are also advertised as solutions, like trusted Transport Layer Security (TLS) certificates with Extended Validation (EV) certificates, but they are too complicated for non-technical users and barely change the outcome. Right now, users lack a way to easily verify that they are using the intended digital service. Verifiable Labels uses cryptographic identifiers—e.g., from the TLS Public Key Infrastructure (PKI)—to bind an entity's label to its identifiable key pair, is a potential solution. Instead of trying to provide automated trust, Verifiable Labels acknowledge the presence of ill-intentioned entities. In order to differentiate them from trustworthy actors, cryptographic tools are used to define metrics, which allow a user client to form easily understandable recommendations and analyze a certain actor's reputation, thus allowing users to naturally develop an opinion and make an educated guess as to whether an entity is trustworthy or not. The end goal would be that most websites asking for some level of trust use Verifiable Labels. This not only has the potential to directly impact Internet users, but also to act as a guiding light for security companies. Since all participating websites would be listed with their reputation metrics, it becomes easier to identify high-risk websites and perform pertinent in-depth analysis in order to take action against phishers faster.**

***Index Terms*—Trust; Anti-Phishing; Digital Label; Reputation.**

I. Introduction

This article presents an extended and refined version of the conference paper titled "Verifiable Labels for Digital Services: A Practical Approach", which was presented during DIGITAL 2023, Advances on Societal Digital Transformation [1].

Nowadays, if website owners want to try and certify an accordance to a label, one sole option is at their disposal: The usage of copyable and thus untrustworthy digital representations, such as pictures or electronic documents, e.g., 'Digital Trust Label' [2]. Without having to make any distinction between true and false claims, it can already be deduced that it has as much value as a self-proclamation and is at least hard and inconvenient, if not impossible, to verify. This is leading naïve Internet users to give their trust to services unworthy of any. Moreover, it is far from affecting only a limited number of people, as since 2020, phishing attacks have become by far the most common type of attacks performed by cybercriminals [3]; 41% of security incidents begin with the initial access gained by a phishing attack [4]; approximately 1.385 million new phishing web pages are set up each month [5]; and overall, phishing is in the top three cybersecurity threat trends [6].

The real problem is there; a verifiable label would truly add value to anybody's Internet experience by directly reducing the impact of phishing. Verifiable labels strive to establish a distributed framework for the development of labels in general and enhance user-friendliness. Additionally, if the concept was successfully adopted, it could act as a guiding light for the existing security ecosystem.

The rest of the paper is structured as follows: Section II analyzes the current state of Internet related technologies; Section III defines the concept of verifiable labels, its underlying infrastructure and protocols; Section IV approaches the concept from a security point of view; Section V explains how the concept was adapted to a working prototype; finally, the work is concluded in Section VI.

II. State of the Art

A. TLS Certificates

Based on Public Key Infrastructure (PKI) to establish chains of trust and using X.509 certificates to bind web-servers to key pairs and domain names, Transport Layer Security (TLS) certificates are nowadays widely used to encrypt communications on the Internet [7]–[9]. These so-called chains of trust are all built upon an entrusted third party—a root of trust—that certifies the trustworthiness of other entities, which in turn are sometimes allowed to do the same. Such entrusted third parties are called Certificate Authorities (CA), as shown in Figure 1.

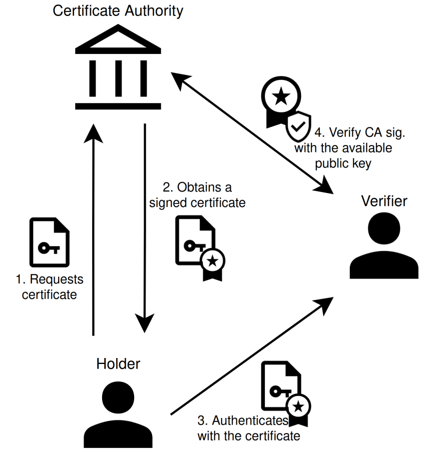

Certificate Authority
4. Verify CA sig. with the available public key
2. Obtains a signed certificate
Verifier
1. Requests certificate
Holder
3. Authenticates with the certificate

Figure 1. Minimalistic representation of a PKI.

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

Furthermore, the X.509 certificate itself can contain a variety of different claims. For instance, one way to bind a certificate to a server is to include its specific domain inside. In the case of TLS certificates, there are three major types of X.509 certificates that are used [10].

a) Domain Validated (DV) Certificate: These are the most basic types of certificates. The CA will only verify that the applicant has control over the requested domain name; this is typically done through email validation. More recently, the Automatic Certificate Management Environment (ACME) protocol allowed CAs to issue DV certificates without any intervention from their side [11]. When the ACME protocol is used, the certificate can be obtained free of charge.

b) Organization Validated (OV) Certificate: Not only is the domain ownership verified, but also the legal existence and physical location of the applicant. Automation is, of course, out of the question. Such a certificate can be obtained for a range of 200 to +1000 USD per year [12].

c) Extended Validation (EV) Certificate: EV certificates undergo the most rigorous validation process; this includes all steps taken for OV certificates, including legal status, operational existence, and telephone verification [13]. The price range goes from 400 to +1700 USD per year [12].

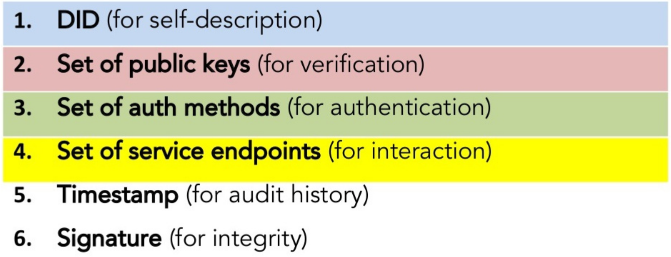
OV and EV certificates were advertised as a way to prevent the customers' users from being prone to phishing, as the web browser, recognizing an EV certificate, used to display a green indicator containing the entity's legal name. Thus, users who knew of that distinction would change their behavior according to the level of certification displayed. However, studies showed that user behavior did not alter [14], and polls [15] showed that the padlock's meaning was not understood correctly. Worse even, security researchers were able to prove that some EV certificates could be gotten with colliding organization names, which could be quite misleading as the domain would be hidden by the legal name in some browsers. That is why, in September 2019, most browsers stopped displaying any direct visual distinctions between DV, OV, or EV certificates, which invalidates the main selling point of these products [16].

Moreover, because CAs are private companies, the regulations are not always followed with the same rigor, as not all validation processes can be automated. A PKI infrastructure is always very sensitive to mistakes, and the verification process has proven to not be enough [17]. However, one thing is sure: TLS certificates do a good job of binding a domain name to its corresponding server, which holds the key pair. Especially with the help of the ACME protocol.

B. Decentralized Identifiers and Key Event Receipt Infrastructure

In opposition to the traditional central authoritative system that CAs and DNS represent, Decentralized IDentifiers (DID) and Key Event Receipt Infrastructure (KERI), both open standards in active development, are part of a broader movement that strives towards decentralized identity.

A Key Event Receipt Infrastructure is a secure and decentralized key management system [18]. It provides mechanisms for proving the Root of Trust for self-certifying identifiers and their associated key states. While TLS certificates bind themselves to a domain name by using trusted third parties, the cryptographic identifiers KERI generates are bound in the strongest manner to their key-pair by using one-way cryptographic functions. This means that the authenticity and integrity of identifiers are verifiable through cryptographic proof. It has great potential and could very well replace the current administrative centralized infrastructure of TLS certificates.



1. **DID** (for self-description)
2. **Set of public keys** (for verification)
3. **Set of auth methods** (for authentication)
4. **Set of service endpoints** (for interaction)
5. **Timestamp** (for audit history)
6. **Signature** (for integrity)

Figure 2. The standard elements of a DID document [20].

A DID resolves to a DID document—typically hosted on a decentralized network or infrastructure, e.g., a blockchain or a distributed ledger—which contains a set of public keys, authentication methods, service endpoints, a time-stamp to keep an audit history, and a signature for its integrity [19].

KERI has already standardized a way to link a KERI identifier to a DID. It leverages KERI's strong cryptographic controls to create decentralized identifiers.

A Verifiable Credential (VC) is a claim created from the key pair of a DID (the issuer) and is issued to a holder's wallet by using a holder proof. This holder proof varies greatly between implementations, and efforts are being made to standardize it. Self-Sovereign Identity (SSI) solutions strive to provide a way to assert, present, and verify claims in a decentralized manner [21].

A verifiable label solution would be quite straight-forward to implement with such technologies. The big issue with them is that not everything is yet standardized. For instance, once a DNS name is linked to a DID, the browsers will not recognize it as trustworthy. The truth is, it is not yet used in practice. For a verifiable label to be used, it needs to work with the current Internet cryptographic tools. It however highlights the need for a solution that adapts to any type of cryptographic identifier.

C. Users awareness

The first thing to identify before designing a solution is what level of awareness users have whilst navigating the Internet and the way cryptographic proofs are naturally understood due to current visual designs. Consider these three types of users.

1) The unaware user
2) The user with no technical background
3) The technically aware user

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

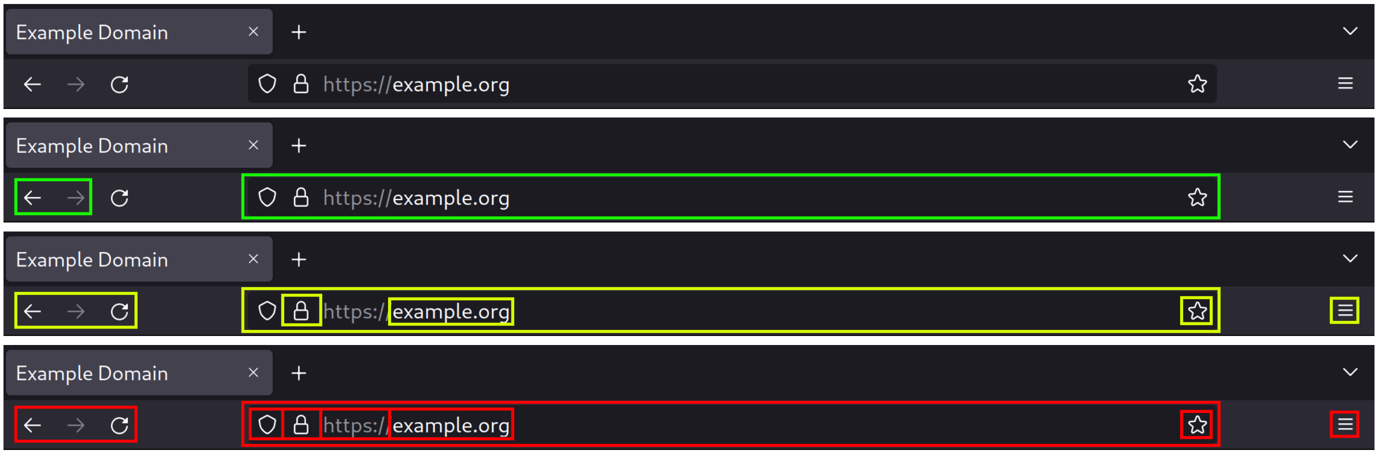


Figure 3. Firefox Browser UI, with each awareness levels depicted from lowest to highest.

These categories were defined after a small survey was conducted. The test consisted in asking users what was their understanding of the purpose of every icon, button or text present on the browser user interface (UI) of Figure 3.

Quick description of the UI:
The back and front arrow navigate through the history, the circle arrow can refresh the page, the shield allows to visualise cookies and trackers present on a website as well as setting protection rules against them, the lock let the user know whether the connection is private or not, `https://` is the protocol, *example.org* is the hostname, the star allows one to add the current URL to their favorites, and the three stacked lines icon is the menu of the browser.

1) Completely unaware users: These users are either very new to the concept of web browsing or rarely use it, they know how to type-in terms, not yet URLs, in the search bar and end up on a page where they can pick a website to visit. They know how to navigate their current session's history with the arrows and, most of the time, they somehow know how to input their credit card's information in any websites. Typically, such users will learn enough to reach the second level of awareness by practicing web browsing. A new solution should ensure that users can intuitively gain awareness as well.

2) Users with no technical background: Most users that often navigate the Internet get to such an understanding of the browser naturally. Through curiosity, they learn how the browser's functionalities work and understand the menus. They know how to add a website to their favorites and have a very basic understanding of a URL. The lock is understood as a 'safe' or 'unsafe' indicator, but there is no comprehension as to why one website would be safer than another. This is a problem, as this indicator does not actually differentiate a spoofer from an authentic website. All it does is indicating whether the connection to the web server is private or public.

3) Technically aware users: Here, users must have wrapped their head around the technical background of the Internet. They know about PKI and SSL/TLS certificates, the HTTPS protocol and how it differs from HTTP and the existence of cookies and trackers. The only way to get there is to study, which is why so many people stagnate in the previous level. But even then, sophisticated spoofing attacks could still succeed in a moment of inattention.

The problem is that SSL/TLS or even HTTPS understanding always remains unknown to the average users, and that the only information they could potentially grasp is a boolean indicator, which is misunderstood and also often misexplained, as 'safe' or 'unsafe'. The plain truth is, no one can ever be a 100% sure that they are visiting the correct website, even with extensive technical verification of the certificate.

If Verifiable Labels were to offer a UI showing green check marks or red warnings beside labels of a website, there would be no major improvements. This is because, although a failed cryptographic proof clearly signals a problem, the successful verification of such a proof does not warranty that no issues are present.

III. CONCEPT

A. Different Perspective

The root of the verifiable label concept lies in a shift of perspective on what trust is and how it can be made identifiable to an end-user. As TLS EV certificates proved, a seemingly good concept will still need to be understood by anyone who uses the Internet in order to have any impact, especially by those who do not have any technical background. First, one must understand how trust is perceived as a concept alone; for this, a philosophical definition of trust is adequate.

'Trust is important, but it is also dangerous. It is important because it allows us to depend on others—for love, for advice, for help with our plumbing, or what have you—especially when we know that no outside force compels them to give us these things. But trust ***also involves the risk*** *that people we trust will not pull through for us, for if there were some guarantee they would pull through, then we would have no need to trust them.* ***Trust is therefore dangerous****. What we risk while trusting is the loss of valuable things that we entrust to others, . . . '* [22]

That is, when a person *decides* to place their trust in someone else, they know about the risks—risks that can be clearly identified as they are based on facts.

Instead of distributing a trust people have to blindly believe in, verifiable labels proposes the idea of providing simple facts about Internet entities so that anyone with no technical background can, in a reasonable time, learn how to navigate the Internet with the ability to discern entities that deserve their trust. To take a risk is, after all, an individual decision, and users must be able to make the assessment themselves and not have to entrust it to a third-party organization that does not have their interests at heart.

To do this, cryptography is paramount, as it is the sole option available to make any virtual information a tangible fact. The system must be implemented on top of the currently widely used Internet cryptographic technologies (e.g., TLS certificates) in order to have any chance of success, while also striving to be flexible and pushing towards more decentralized technologies (e.g., blockchains) because they provide a better and stronger infrastructure.

B. Definitions

1) VERIFIABLE LABEL
   A verifiable label is a data structure that is bound to two domain names; the holder's and the issuer's. This is done by signing the label with both domain name linked cryptographic identifiers (e.g., TLS certificate). This ensures that the label is bound to the server that holds the cryptographic identifiers. Therefore, ensuring the authenticity of both the holder and issuer, and that it cannot be copied. It also warrants the integrity of the content. It contains the following fields:
   a) Domain of holder
   b) Label name
   c) Domain of issuer
   d) Signature of the holder's cryptographic identifier
   e) Signature of the issuer's cryptographic identifier
2) ISSUER RECORDS
   An issuer record is another data structure similar to a verifiable label, it defines the identity of an issuer and its label while also fulfilling a variety of other roles. Since its validity is warranted by an external entity's—i.e., the verifiable enforcer's—signature, each time a holder is added or removed from the record, a new one has to be requested if an immediate update is deemed necessary. On top of that, the system will ensure that it has to be renewed once its time-stamp is too old. All previous records are to be kept online by the issuer, for each of them contains valuable behavioral data when combined with the rest. To ensure that all are present, each of them is bound by the previous record's signature of the verifiable enforcer. It contains the following fields:
   a) Label name
   b) Domain of issuer
   c) List of holders [(Domain of holder; Status; Issuer signature)]
   d) Previous verifiable enforcer's signature
   e) Timestamp
   f) Signature of the issuer's cryptographic identifier
   g) Signature of the verifiable enforcer's cryptographic identifier
3) VERIFYING USERS
   Simple users that visit a website. If a valid label is detected, the user will be able to see it, list facts that concern it, and develop an idea of this label's reputation.
4) LABEL-WORTHY WEB-ENTITY
   Such an entity can request a label from its corresponding issuer. If an issuance occurs, it officially becomes a holder and can display its digital label on their website, which is visible and verifiable by anyone. If copied to an alternative server and a different TLS certificate is used, validation will always fail.
5) ISSUER
   The entity that can verify and decide, of its own accord, who is worthy of being labeled. It will keep a record of who has been issued its label and can confirm it with the list of holders contained in its latest record. This gives the issuer the power to revoke a label by just removing the entry from the list.
6) VERIFIABLE ENFORCER
   The backbone of the concept is here; this entity can be understood both as a centralized, transparent authority, or as a set of rules enforced by a consensus. As it is the only piece of infrastructure that would require financing if implemented centrally, a decentralized implementation would be preferable, i.e., as a smart-contract [23]. It will follow specific automated guidelines, all of which are reproducible and thus verifiable. It will provide time-stamps from a trusted source—either a time-stamp authority or a blockchain time-stamping service [24]—and distribute them with a signature to pre-existing issuers on demand. As all issuers need an unexpired latest record, they will have to issue renewal requests in any case. Each request, building upon one another, starts to create a reputation. The purposes of the guidelines are to make sure that every issuer plays by the same rules by recording each of their behaviors in their own records, as well as to aim at enforcing duplicate label prevention when a new label requests its first signed record. The only data necessary for it to operate is all of the issuer's domains, which is easy enough for it to keep track of. This list will be publicly available but will have no other purpose than allowing exploration.

C. Protocol

a) Issuance of a label: Figure 4 depicts it. A website must create a verifiable label and sign it with its TLS certificate. This ensures that the draft label is bound to the domain name and also comes from the stated owner. The incomplete digital label can be sent to the issuer; no channel is specified. If the issuer decides to accept the request, it will sign it with its own TLS certificate, add the new signature to the now complete verifiable label, and send it back. Finally, the issuer save a copy of the signature and requester's domain in the

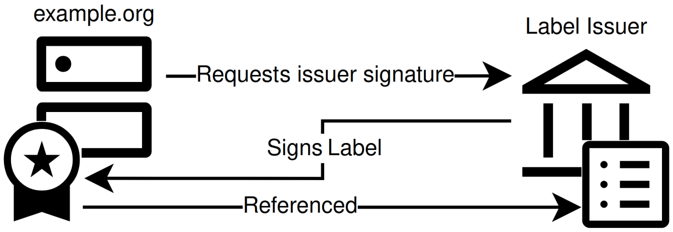


Figure 4. Verifiable Label Issuance

list of its new draft record. In order to make a valid issuer record out of this draft, the issuer has to request a new time-stamp and signature from the verifiable enforcer, as explained in Figure 5.

b) Issuer Record Validity: As stated before, an issuer's trustworthiness is defined by its own reputation. This reputation is built with time and the help of a trusted time-stamp source. The verifiable enforcer's role is to issue new signatures—necessary for the issuer's label to be considered cryptographically valid—and time-stamps to all pre-existing requesting issuers that are on the brink of expiration. As it does so, each request will always be examined in a replicable manner that warrants the verifiable enforcer's verifiability. First, it will verify the cryptographic signature of the new incomplete record and its previous records, which must be kept online. Once authenticity has been verified, each holder's verifiable label is retrieved from the list contained in the draft record, and each of their status field will be set to a boolean value that reflects whether or not said website is online with its verifiable label. Finally, the draft is signed and then sent back. However, if the issuer is new, i.e., does not possess a previous signature, the verifiable enforcer will take a look at the requested label name, domain name, and all fields that might be prone to confusing a human being. If it is considered not to be confusing as well as not a duplicate of any existing labels, the web entity will receive its first time-stamp and signature, making it an issuer.

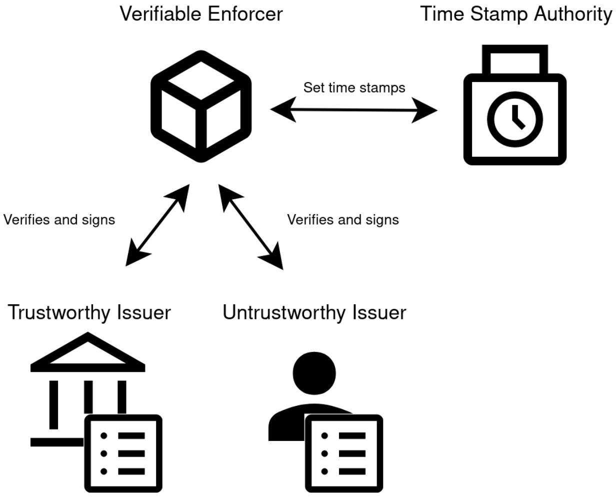


Figure 5. Verifiable Enforcer

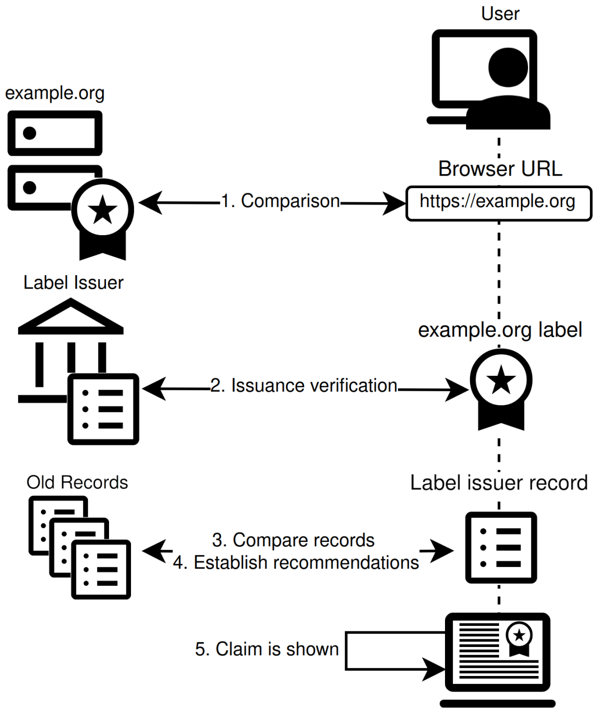


Figure 6. Verifiable Label Validation

c) Validation and interpretation client: As a user with the verifiable label validation and interpretation client installed navigates the Internet, the client will try to detect if a digital label is present on the currently visited website. If this proves to be the case, the validation process will begin, as shown in Figure 6. The first step consists of verifying the label's link with the domain and TLS certificate, that is, making sure the signature is correct and that the domain corresponds to the browser URL. On success, the next step will be to cross-check the label with the listed issuer record. The record and digital label are to be compared, signatures are to be verified for the same reason as before, and the domain of the holder must be found in the list. At last, if everything succeeds again, the client will go through the label's previous' records to derive a reputation.

D. Reputation System

Accurate protection is possible if we assume that a majority of web entities have adopted this digital label system. Only then would websites with bad or no label start to stand out, especially because they require trust, e.g., when they ask for credit card information or propose services.

IV. SECURITY CONSIDERATIONS

Phishing attacks have been studied for years now. Some recent scientific papers as well as older ones [25]–[27] have described how a typical attack is performed, as shown in Figure 7. Most of the security industry focuses on stopping the attack at step 2. The techniques used can be on the client side, like native browser blacklists and malware detection tools that analyze URL and page content or search engine rankings;

Whereas on the server side, the user would have to check for an EV certificate, i.e., if he knows about it and if the website still pays for one. Furthermore, security companies use a network of systems to track and update blacklists [27].

Thus, the anti-phishing ecosystem can be described as reactive, and a requirement for a reactive system to be effective is at least one low latency feedback source. However, the sources used by the current ecosystem seem to always be a few steps too far behind to provide effective measures at the right time.

Under the assumption that Verifiable Labels was put in place successfully, most business websites (holders) would have at least been issued one label to assert their online identity. Since every issuer of label is being accounted for, as is each of its holders' websites, it ultimately creates a list of links ready to be scanned. As a result, the set of websites to be analyzed is reduced to this list, which contains all websites that require the trust of their users, for that is where phishers want to be.

Therefore, when phishers eventually change their strategy to try and fulfill this new requirement, they will face a bigger risk of being discovered, as security companies will be able to actively search for them in a defined subset.

In an effort to already foresee and impede an attacker's effort to try and infiltrate the Verifiable Label system, further measures were imagined and are described in the sections below.

A. Issuer Minimal Requirements

Since phishers leverage the existing trust a user gives to an existing company by impersonating it, the following measures will try to hinder the creation of duplicate labeling companies.

Before a labeling organization can start issuing labels, as stated in the protocol subsection III-C0b, its new label will undergo a replicable analysis conducted by the verifiable enforcer:

1) With the given signature and the corresponding TLS certificate, the domain of the issuer is authenticated.
2) The domain's first prefix, as well as the label name, have to be unique. Moreover, the remaining suffix of the domain has to be part of a whitelist.
3) It is yet undecided whether a supervised machine learning clustering algorithm or a more traditional algorithm should strive to eliminate new labels with names and domains that resemble other well-known issuers or are considered confusing.

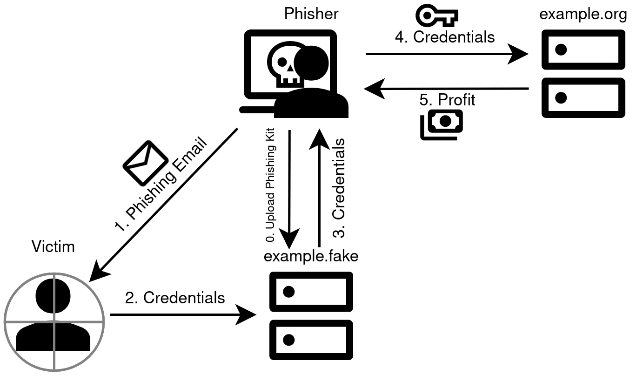


Figure 7. Typical Phishing Attack

As stated above, this process is reproducible and thus verifiable as well. It is to be applied uniformly in an automated manner that allows others to come to the same conclusion by reproducing the analysis. This should already impact the way impersonations are engineered when trying to create a fake label, thus reducing the potential risk.

B. Monitoring & Pattern recognition

To assume that a new issuer is trustworthy once it passes the initial registration verifications, as older systems showed, is wrong. Phishers will adapt. And so should the infrastructure supposed to safeguard the Internet trust. Verifiable Labels proposes a new approach to phishing detection: to provide recent data that characterizes the behavior of issuers, including their holders, and leverage it to evaluate the risk of trusting each entity.

What is evaluated is not trust, because, as demonstrated before, the end-user should always remain in control and choose for himself with the help of facts, i.e., the metrics in our case. However, a client-side risk-based evaluation could help further. This way, a user not only gets additional knowledge about the behavior of the label of each website, but also an illustrated meaning. Figure 8 clarifies each actor's domain of action and/or responsibilities in the Verifiable Labels ecosystem.

Please note that the weights applied to the following metrics are only meant to serve a temporary purpose while historical data is not available. A machine learning algorithm could be much more effective at identifying such weights, as an ever-growing historical dataset would allow it to adapt to the new trends and immediately fight back.

a) Age: The age of a specific label is quite different from a domain's age; research shows that 53.3% of phishing attacks used domains older than a year, it is generally assumed that this is because phishers were able to use compromised infrastructure in their campaigns [27] . Since a label is strongly bound to the cryptographic key of the server, the attacker also has to compromise the cryptographic identifier to take over a label issuer, making it harder, or close to impossible if the cryptographic identifier is well managed with KERI [18]. Hence, a new label would have a bigger risk of being fake than an old one.

Consequently, age will impact the risk the most when, e.g., a label is younger than a year. In this manner, all new labels would stand out.

b) Number of Holders: The second metric is the number of holders an issuer has. It can be counted by any client from the list of holders contained in the latest issuer's record. A low amount does not make sense and is easy for a phisher

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

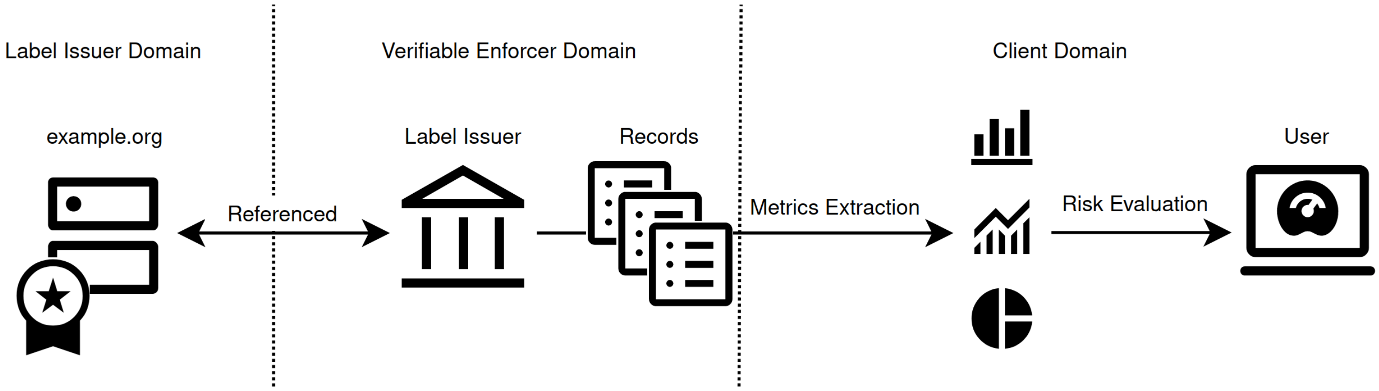


Figure 8. Delimitation of Each Actor's Domain

to achieve, while on the other hand, having many holders is natural and is quite hard for a phisher to obtain and maintain. It should pressure the phishers to group, which increases their infrastructure costs but also potentially attracts unwanted attention.

c) Number of Dead Holders: Holders that could not be reached by the verifiable enforcer during issuance were listed in the issuer record. It will penalize the issuer's risk evaluation heavily if any is present. In this manner, issuers are incentivized to make sure that they not only have a good number of holders, but also that the holders are qualitative. Therefore, making sure that the list of holders is up-to-date and that phishers are on their own.

d) Attrition Rate Analysis: Inspired by the concept of churn rate analysis, but with identifiers and a different purpose.

A client will identify all holders that were removed by going back through the old records of a label. All of these deleted holders will have their lifetime calculated; a long average lifetime is expected; otherwise, it would become clear that a portion of holders are being constantly replaced in a devious attempt to evade detection. The number of removed holders is also expected to be very low in comparison to the total number of holders.

Needless to say, a client will give much weight to both values in the risk evaluation.

e) Surge Detection: A surge is when a large number of holders are either added or removed. This mechanism is very useful to forbid and punish sudden increases in a number of holders, or a sudden evasion technique that consists in removing all phishers. The impact on the evaluation is to be proportional to the spike's intensity, making it temporarily stand out. However, with time, the burden will lessen.

f) User Feedback: Should all of the measures still fail to identify a malevolent entity, clients will be equipped with a reporting functionality that send all parameters that are used in the context of its visit—i.e., the user-agent, the holder's website, and a reason—in order to avoid cloaking mechanisms on the targeted website. Each time a report is sent, the user has to prove that they are human by filling out a captcha. Such feedback reports will be stored and made publicly available by the verifiable enforcer, and will not have any weight in the evaluation of risk. However, it should provide insights as to what subset of websites might be worth scanning with the reported parameters.

C. *Integration with Existing Security Infrastructures*

Hypothetically, if a web crawler was ordered to visit and evaluate the set of websites corresponding to all web-entities that use Verifiable Labels, it could filter them into an even smaller subset corresponding to all issuers, including their holders, with a medium to bad reputation.

This could greatly benefit the existing security infrastructure by pointing to the websites that require urgent measures, thereby rapidly winning more crucial battles against phishers. Which, in turn, will effectively lower the latency of feedback central to the reactive security ecosystem.

V. Implementation

A prototype has been implemented following a minimal working system approach. Furthermore, since different underlying technologies exist, extensibility is a top priority.

A. *Verifiable Enforcer*

Starting from the very root of the system, this implementation of verifiable enforcer uses a library that implements an RFC 3161 [28] client interface to interact with an external TSA to provide the time-stamps. A TLS certificate was used to sign issuer records. This server software consists of a simple HTTP API with two paths: the POST method on '/sign' and the GET method on '/get_records'. Meaning, it also acts as a publicly readable storage. All of this has been implemented in the most minimalistic way, with abstract interfaces of 'Storage', 'API', and 'Signer'. That is where flexibility is; the logical part of what makes the verifiable enforcer is detached from all other components that could find better long-term alternatives (e.g., more resource-efficient or different time-stamp sources such as a blockchain).

B. Issuer Client

The simple command-line client has persistent storage and saves all valid given arguments. If provided with a valid request, it will add a domain to its record and generate a valid digital label (.vlcert), which can be sent back to the holder through any channel. It can issue a signing request to the verifiable enforcer on demand. And, if successful, it will save the verifiable label issuer record (.vlicert). All vlicert have to be exposed on the label domain's web server root.

C. Holder Client

This simple command-line client with no persistent storage can only be used to generate a verifiable label without the issuer signature. It has to be manually sent to the issuer. Once a valid digital label (.vlcert) is in the holder's possession, it has to be exposed on its domain's web-server root as 'cert.vlcert'. This prototype thus only allows for one label per holder.

D. Browser Extension Analyzer

A browser extension was a mandatory component of the client, as the active URL has to be accessed to perform the first cryptographic tests. However, the specific environment did not provide any way to download a TLS certificate for a specified domain, which blocked further development. More research showed that by using the native messaging interface, the browser extension can communicate data to an underlying program. Using this method, a cryptographic verifier was developed. It sends back the necessary data to perform a reputation analysis and is then displayed in a panel.

VI. Conclusion

We proposed a system to implement a reliable reputation-based digital label system for websites to replace the now fragile and ineffective automated trust provided by the current security infrastructure.

Each website can request labels from self-declared label issuers. Each issuer, whether trustworthy or not, has its label activities monitored and stored in its own issuer records. This is ensured through the verifiable enforcer, whose timestamps and signatures are necessary for an issuer to be recognized as such. All records of each issuer are always kept online, allowing client software to extract pertinent metrics from them and evaluate the overall reputation of a label through a risk analysis. This risk analysis should allow humans to develop a sense of trustworthiness without having to understand Internet-related technologies.

It was argued that verifiable labels would mostly be used by websites that require the trust of their user base (e.g., webshops require credit card information). And since phishers leverage the pre-existing trust between a user and a web entity, it would only be a matter of time before they tried to infiltrate the reputation system. The metrics provided in every issuer record are thus not only useful to evaluate the risk for a user, but they would also be very pertinent to identifying what subset of websites should be on the watch list of the current security industry.

Furthermore, a minimalistic prototype was implemented. It is flexible, and, even if simplistic, it already implements all the necessary cryptographic tools.

Future work could investigate the following directions:

- Extend the prototype to support fully decentralized infrastructures.
- Conduct a field study of a live setup and user experience.
- Prove the effectiveness of the reputation metrics.
- Provide a comprehensive User Interface (UI) for computers and phones.

References

[1] M. Gassmann and A. Laube, "Verifiable labels for digital services: A practical approach," in *DIGITAL 2023, Advances on Societal Digital Transformation*. Wilmington, DE, USA: International Academy, Research, and Industry Association (IARIA), 2023, pp. 8–12, [retrieved: 03, 2024]. [Online]. Available: https://www.thinkmind.org/index.php?view=article&articleid=digital_2023_1_20_20012

[2] Swiss Digital Initiative. (2024) The digital trust label. [retrieved: 05, 2024]. [Online]. Available: https://digitaltrust-label.swiss/

[3] FBI Internet Crime Complaint Center, "Internet crime report 2023," Federal Bureau of Investigation, Tech. Rep., 2023, [retrieved: 05, 2024]. [Online]. Available: https://www.ic3.gov/Media/PDF/AnnualReport/2023_IC3Report.pdf

[4] IBM Security, "X-Force Threat Intelligence Index 2023," International Business Machines Corporation, Tech. Rep., 2023, [retrieved: 08, 2023]. [Online]. Available: https://www.ibm.com/reports/threat-intelligence

[5] M. Swindells. (2023) How many phishing emails are sent daily in 2023? 11+ statistics. [retrieved: 08, 2023]. [Online]. Available: https://earthweb.com/how-many-phishing-emails-are-sent-daily/

[6] Cisco Umbrella, "Cybersecurity threat trends: phishing, crypto top the list," Cisco, Tech. Rep., 2021, [retrieved: 08, 2023]. [Online]. Available: https://umbrella.cisco.com/info/2021-cyber-security-threat-trends-phishing-crypto-top-the-list

[7] E. Rescorla and T. Dierks, "The Transport Layer Security (TLS) Protocol Version 1.2," RFC 5246, Aug. 2008, [retrieved: 08, 2023]. [Online]. Available: https://www.rfc-editor.org/info/rfc5246

[8] E. Rescorla, "The Transport Layer Security (TLS) Protocol Version 1.3," RFC 8446, Aug. 2018, [retrieved: 08, 2023]. [Online]. Available: https://www.rfc-editor.org/info/rfc8446

[9] R. Housley, T. Polk, D. W. S. Ford, and D. Solo, "Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile," RFC 3280, May 2002, [retrieved: 08, 2023]. [Online]. Available: https://www.rfc-editor.org/info/rfc3280

[10] CA/Browser Forum, "Baseline requirements for the issuance and management of publicly-trusted tls server certificates, version 2.0.2," January 2024, [retrieved: 01, 2024]. [Online]. Available: https://cabforum.org/uploads/CA-Browser-Forum-TLS-BRs-v2.0.2.pdf

[11] R. Barnes, J. Hoffman-Andrews, D. McCarney, and J. Kasten, "Automatic Certificate Management Environment (ACME)," RFC 8555, Mar. 2019, [retrieved: 08, 2023]. [Online]. Available: https://www.rfc-editor.org/info/rfc8555

[12] DigiCert. (2023) Compare tls/ssl certificates. [retrieved: 08, 2023]. [Online]. Available: https://www.digicert.com/tls-ssl/compare-certificates

[13] CA/Browser Forum. (2022) Ev ssl certificate guidelines. [retrieved: 08, 2023]. [Online]. Available: https://cabforum.org/extended-validation/

[14] C. Thompson, M. Shelton, E. Stark, M. Walker, E. Schechter, and A. P. Felt, "The web's identity crisis: understanding the effectiveness of website identity indicators," in *28th USENIX Security Symposium (USENIX Security 19)*, 2019, pp. 1715–1732.

[15] PhishLabs. (2017) A quarter of phishing attacks are now hosted on https domains: Why? [retrieved: 03, 2024]. [Online]. Available: https://www.phishlabs.com/blog/quarter-phishing-attacks-hosted-https-domains

[16] Chromium Docs. (2019) Ev ui moving to page info. [retrieved: 08, 2023]. [Online]. Available: https://chromium.googlesource.com/chromium/src/+/HEAD/docs/security/ev-to-page-info.md

[17] C. Cimpanu. (2017) Extended validation (ev) certificates abused to create insanely believable phishing sites. [retrieved: 08, 2023]. [Online]. Available: https://www.bleepingcomputer.com/news/security/extended-validation-ev-certificates-abused-to-create-insanely-believable-phishing-sites

[18] S. M. Smith, "Key event receipt infrastructure (KERI)," *CoRR*, vol. abs/1907.02143, 2019, [retrieved: 11, 2023]. [Online]. Available: http://arxiv.org/abs/1907.02143

[19] D. Reed, M. Sabadello, A. Guy, and M. Sporny, "Decentralized identifiers (DIDs) v1.0," W3C, W3C Recommendation, Jul. 2022, https://www.w3.org/TR/2022/REC-did-core-20220719/.

[20] D. Reed. (2018) Decentralized identifiers (dids): The fundamental building block of self-sovereign identity (ssi). [retrieved: 11, 2023]. [Online]. Available: https://www.slideshare.net/SSIMeetup/decentralized-identifiers-dids-the-fundamental-building-block-of-selfsovereign-identity-ssi

[21] D. Longley, D. Burnett, K. D. Hartog, M. Sporny, B. Zundel, and G. Noble, "Verifiable credentials data model v1.1," W3C, W3C Recommendation, Mar. 2022, https://www.w3.org/TR/2022/REC-vc-data-model-20220303/.

[22] C. McLeod, "Trust," in *The Stanford Encyclopedia of Philosophy*, Fall 2021 ed., E. N. Zalta and U. Nodelman, Eds. Metaphysics Research Lab, Stanford University, 2021, [retrieved: 08, 2023].

[23] V. Buterin, "A next-generation smart contract & decentralized application platform," https://ethereum.org/en/whitepaper/, Ethereum Foundation, Tech. Rep., 2014, [retrieved: 05, 2024].

[24] L. Meng and L. Chen, "A blockchain-based long-term time-stamping scheme," in *Computer Security – ESORICS 2022*, V. Atluri, R. Di Pietro, C. D. Jensen, and W. Meng, Eds. Cham: Springer International Publishing, 2022, pp. 3–24, [retrieved: 01, 2024].

[25] D. Watson, T. Holz, and S. Mueller, "Know your enemy: Phishing - background on phishing attacks," https://honeynet.onofri.org/papers/phishing/details/phishing-background.html, The Honeynet Project & Research Alliance, Tech. Rep., 2005, [retrieved: 11, 2023].

[26] P. J. Nero, B. Wardman, H. Copes, and G. Warner, "Phishing: Crime that pays," in *2011 eCrime Researchers Summit*, 2011, pp. 1–10, [retrieved: 11, 2023].

[27] A. Oest, Y. Safei, A. Doupé, G.-J. Ahn, B. Wardman, and G. Warner, "Inside a phisher's mind: Understanding the anti-phishing ecosystem through phishing kit analysis," in *2018 APWG Symposium on Electronic Crime Research (eCrime)*, 2018, pp. 1–12, [retrieved: 11, 2023].

[28] R. Zuccherato, P. Cain, D. C. Adams, and D. Pinkas, "Internet X.509 Public Key Infrastructure Time-Stamp Protocol (TSP)," RFC 3161, Aug. 2001, [retrieved: 08, 2023]. [Online]. Available: https://www.rfc-editor.org/info/rfc3161

A Multivocal Review on Derivation Games: A Software Reuse Study

Diego Castro*†, Claudia Werner*
*Program Systems Engineering and Computer Science - COPPE
Federal University of Rio de Janeiro
Rio de Janeiro, Brazil
†Federal Center for Technological Education Celso Suckow da Fonseca (Cefet/RJ)
Rio de Janeiro, Brazil
email: {diegocbcastro, werner}@cos.ufrj.br

Abstract—**Games have emerged as a prominent form of entertainment, hence establishing the gaming business as a highly lucrative sector. Nevertheless, the process of developing a game can be extremely complex, involving several activities, components, and team members, which can result in an extremely long development period for certain games. The gaming community is engaging in constructing its own games as a response to the delay; its behavior is similar to opportunistic Software Reuse (SR), which is also called modding. The application of mods in game development can provide several benefits, including enhanced longevity of games, reduced production expenses, and accelerated creation of diverse games within reduced timelines. However, the current mod development process lacks a systematic framework or platform that can assist in this progress. Consequently, the aim of this research is to conduct an evaluation that clarifies the primary characteristics, advantages, challenges, and approaches employed in the development of mods, with the goal of comparing these findings with the RS methods that are recommended by the market.**

Keywords-Software Reuse; Game; Mods; Derivation; Structured review; Multivocal review.

I. INTRODUCTION

This work is an extension of the paper presented by Castro and Werner [1] at the 2023 IARIA Annual Congress on Frontiers in Science, Technology, Services, and Applications (IARIA), Valencia, Spain.

The evolution of games has been remarkable, establishing itself as one of the most prevalent kinds of entertainment in modern times. This industry generates billions of dollars annually in revenues and investments [2, 3]. As a result of its exponential development over time, this sector has amassed a substantial user community, attracting people of all genders and ages. However, despite the substantial expansion and the extensive fan base, the game development procedure has been exhaustive and deficient in organization, resulting in delays for several titles' release dates [4]. This delay in launching a certain game could cause anxiety and irritation in this community, contributing to search for new games or even their creation of games [5].

By doing a brief search, it is possible to identify several websites that offer modified versions of games. A modification, also referred to as a "mod," involves one or several alterations or adjustments made to a game, which could be related to its mechanics, dynamics, or any other basic element. Its classification may differ contingent upon the degree of modification: these categorizations involve terms such as patches, tweaks, add-ons, and other designations [6, 7, 5]. This method of modifying games can result in a variety of advantages for the company that created the original titles. Among the primary advantages there are: an increase in the number of users, the number of sales, and the longevity of the game [2].

The development of mods carried out by the gaming community can be directly compared to the opportunistic Software Reuse (SR) approach, where a user reuses something previously created to develop new software. Therefore, the objective of this study is to review existing literature to determine the most common ways of mod development currently in use, in order to improve these practices using recommended SR methodologies. As previously mentioned, the gaming community has a substantial user base. Therefore, the search methodology employed in this study aimed to gather material from both formal research sources and informal channels, taking into account the vast size of this community. Due to this, the research method used was Multivocal review which incorporates data from both white (academic papers, books, etc.) and gray (blogs, websites, videos, etc.) literature [8]. This strategy is typically utilized when there is substantial community support for the study subject and it is necessary to verify practical knowledge on a particular subject

The subsequent sections of this paper are outlined as follows: Section II provides a concise overview of the research procedure used in this study. Section III includes a comprehensive analysis of the data encountered during the search process. Finally, Section IV offers a concluding summary of the paper.

II. THEORETICAL FOUNDATION

A. Games

Multiple definitions of games can be found in the literature. Most of them, however, revolve around the following definition: Games can be defined as activities that utilize a conceptual setting where choices, actions, and rules are established in order to engage in a recreational effort, such as entertainment or pleasure [9, 10].

Identifying the defining characteristics of a new game is a more complex task than it may initially seem. According to the previous definition, it is possible to identify some characteristics that differentiate them, including new rules, actions and choices made by players. Therefore, a mod can be

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

identified as a game that has received certain modifications in some of these previously mentioned characteristics [6, 7].

B. Software Reuse

The practice of reuse is widely employed across various industries, such as manufacturing, automotive, and electronics. The term Software Reuse (SR) was first introduced at a NATO conference in 1968 [11] and is one of the disciplines of Software Engineering (SE). Software reuse is the practice of creating new systems by utilizing existing software artifacts and knowledge, rather than building everything from scratch [12].

SR has several areas of study within its field of research, such as: component-based development, Model Driven Development (MDD) and Software Product Line (SPL).

- **SPL:** is a set of strategies, techniques, and tools used to systematically develop similar systems that share a common core but have unique features. The application of these subjects is expected to lead to a reduction in development time, easier maintenance and evolution of systems, increased programmer satisfaction, and improved code quality [12].
- **Software Component:** can be conceptualized as an autonomous and interchangeable piece of code that fulfills a specific function and is reused during the development of a new program. By implementing this approach, several benefits are expected to be achieved, such as the unification of functions, encapsulation, improved code quality, and more agile development [13].
- **MDD:** aims to separate feature design from implementation specification. This technique facilitates software development by utilizing modeling and implementing models of its implementations. The primary objective of this method is to allow developers to concentrate solely on the application's business specifications, rather than the underlying platform on which it will run [14].

III. RESEARCH PROTOCOL

As mentioned, a Multivocal Review (MR) is a more complete examination of the literature that aims to elicit as much information as possible about a specific subject; hence, it incorporates data from both white (academic papers, books, etc) and gray (blogs, websites, videos, etc) sources. A MR may be separated into two stages: the first stage involves the search for academic knowledge (in this case, a Mapping Literature Review - MLR - was used), and the second stage involves the search for gray literature [8]. MLR is a systematic methodological review of research that investigates and categorizes studies in a specific field of study and presents an overview of a certain subject systematically [15].

In the initial phase of the investigation, four search databases were utilized, following the recommendation of Kitchenham et al. [15]. The search string was executed on the main search engines:

1) **Scopus:** www.scopus.com
2) **ScienceDirect:** www.sciencedirect.com
3) **IEEEXplore:** www.ieeexplore.ieee.org
4) **EI Compendex:** www.engineeringvillage.com

In order to facilitate the execution of this study, a fundamental search string was formulated based on the PICOC framework, which covers the following components: Population, Intervention, Comparison, Outcome, and Context [16]. Combining domain-specific keywords with the logical operator ”OR” and fields with the logical operator ”AND” produced the search string. This string was utilized for the duration of the search. To validate the search string, two control papers (Modding as part of game culture [5], Serious mods: A case for modding in serious games pedagogy [17]) were used to generate and execute the string in the Scopus database, the first database to which the string was applied. This validation technique seeks to ensure the quality of the search string by returning only relevant articles and author knowledge [18].

According to [19] and [20], snowballing processes can compensate for the absence of other search engines and supplement the approach by doing research via the references and citations of the papers. Therefore, to minimize the loss of some papers and increase the search range, the forward and backward (one-level) snowballing procedure was used, which checks the references and citations of articles seeking relevance [21]. The procedures, inclusion and exclusion criteria and quality criteria will be described below. The research questions of each of the reviews will be described in the following sections.

1) **Snowballing Backward**: refers to the identification of new papers based on the works that were referenced in the paper that was analyzed [21];
2) **Snowballing Forward**: refers to the identification of new papers based on the works that referenced the paper that was analyzed [21].

The research execution procedure consisted of the following steps:

1) Execute the search string. For searches in gray literature, it was searched for each search string up to page 10 of google. The search strings were formed by combining the keywords of population and intervention;
2) Apply the inclusion / exclusion criteria based on the title;
3) Apply the inclusion / exclusion criteria based on the abstract;
4) Apply the inclusion / exclusion criteria based on the full text;
5) Apply the quality criteria; Apply snowballing backward;
6) Apply snowballing forward. For searches in gray literature, the snowballing was performed on site references, on links contained within the site.

The inclusion criteria, exclusion criteria, quality criteria, and research questions used in the study were:

Inclusion Criteria:

- Viability Study: The document must be in the context of Mods;

Table I
ANALYSIS OF THE PAPERS ABOUT MUTATORS AND GAMES (MAIN STUDY).

	Scopus		**ScienceDirect**		**IEEEXplore**		**EI Compendex**	
Activity	**Result**	**Number of paper**	**Result**	**Number of paper**	**Result**	**Number of paper**	**Result**	**Number of paper**
First Execution	512 added	512	101 added	101	183 added	183	127 added	127
Number of papers	**1440 papers**							
Repeated Papers	51 withdraw	461	7 withdraw	94	23 withdraw	160	0 withdraw	127
Remove conference	48 withdraw	413	7 withdraw	87	0 withdraw	160	0 withdraw	127
Papers in another language	10 withdraw	403	0 withdraw	87	0 withdraw	160	0 withdraw	127
Number of papers	**1110 papers**							
Remove by title	324 withdraw	78	77 withdraw	10	139 withdraw	21	96 withdraw	36
Number of papers	**207 papers**							
Remove by abstract	59 withdraw	14	10 withdraw	0	18 withdraw	3	20 withdraw	16
Number of papers	**56 papers**							
Papers not found	0 withdraw	14	0 withdraw	0	0 withdraw	3	0 withdraw	16
Remove by quality criteria	2 withdraw	12	0 withdraw	0	0 withdraw	3	6 withdraw	10
Remove by full paper	0 withdraw	12	0 withdraw	0	1 withdraw	2	10 withdraw	0
Extracted Papers	**14 papers**							

Table II
SEARCH STRING OF MUTATOR AND GAMES.

P	*Game*
I	Mutator, variant, mods, modification, conversion, add-on, tweak, modding
C	Not applicable
O	Tools, approach*, method*, ideas, framework*, mechanics, interpretation*
C	Creation, production, development, elaboration, generation, practice*
(*game*) AND (mutator OR variant OR mods OR modification OR conversion OR add-on OR tweak OR modding) AND (tools OR approach* OR method* OR ideas OR framework* OR mechanics OR interpretation*) AND (creation OR production OR development OR elaboration OR generation OR practice)) AND (LIMIT-TO (SUBJAREA, ”COMP”) OR LIMIT-TO (SUBJAREA , ”ENGI”))}	

- The document must be in the context of Games and Software Reuse;
- The document must provide data to answer at least one of the research questions;
- The paper must be written in English.

Exclusion Criteria:

- Conference call;
- Studies that can not be fully accessed;
- tudies that are not in the area of Computer Science or Engineering.

Quality Criteria:

The quality criteria employed are derived from Lincoln and Guba, with the objective to evaluate the author’s credibility, the transferability of ideas to the new paper, the reliability of the information, and the confirmability of the information [22].

- Is the publishing organization reputable?
- Has the author published another work in the area?
- Does the author have expertise in the area?

- Is the article clear?
- Are the references documented?
- Does this enrich the research?

Research Questions:

- **Q1:** What modifiers are used to create games from other games?
- **Q2:** What characteristics are needed to derive a game?
- **Q3:** What are the advantages and difficulties of creating games from others?
- **Q4:** What tools strategy or frameworks support these changes?

The first stage returned a total of 923 papers. When the publications were examined using the inclusion and exclusion criteria, this number was reduced to 14. From these studies, the snowballing process was carried out, and a total of 245 more papers were evaluated. After this approach, 9 more papers were included, totaling 23 papers read and assessed. Tables II, I and III show the search string and the papers' analysis process. Figure 1 demonstrates the steps that were taken when performing the search. It is worth remembering that the paper analyses were broken into two tables, one for the main study and another for the snowball processing.

Based on the findings of the initial phase, it was determined that the gaming community is quite active in terms of development, enhancements, and modifications. Consequently, a new phase was introduced to the study. In addition to the investigation, a search for gray literature was conducted.

The gray literature search encompassed up to page 10 of Google for each of the search keywords, resulting in 700 links that required validation. The inclusion and exclusion criteria were implemented after visiting each link, resulting in the selection of 21 links for the quality criteria step. Ten links were selected and approved based on the following criteria. The snowball effect was achieved by utilizing backlinks (website reference connections). As a consequence, the entire procedure was restarted for the authorized connections, and 335 additional links were validated. Lastly, 12 additional documents were added in the search at the snowballing process. The papers that were chosen for this research are listed in Table IV, along with the questions that each document may answer.

IV. RESULTS ANALYSIS

The rise of the mod trend is closely related to increased accessibility to personal computers and the expansion of the internet, which is disseminating an increasing amount of content [23]. The community and academy are increasingly generating game adaptations, assisting game producers in a variety of ways, including recruiting new players, extending the life of a game, providing new views for the game, and resolving bugs. Modifications, in general, are referred to as mods and may be thought of as modifications to an original game [6].

Modifications to products in the gaming industry done by gamers are now often referred to as modding. Modders employ a variety of strategies in their creations, ranging from basic rearranging of game world parts to complete conversions that can be somewhat independent of the original game [23]. This section will discuss the many sorts of modifiers discovered, their benefits and drawbacks, and lastly, the essential criteria for constructing an adapted game, as well as if there are tools available to aid in this process.

Table III
ANALYSIS OF THE PAPERS ABOUT MUTATORS AND GAMES (SNOWBALLING STUDY).

	Snowballing Backward		**Snowballing Forward**	
Activity	**Result**	**Number of paper**	**Result**	**Number of paper**
First Execution	381 added	381	136 added	136
Number of papers	**1440 papers**			
Repeated Papers	141 withdraw	240	20 withdraw	116
Remove conference	0 withdraw	240	0 withdraw	116
Papers in another language	4 withdraw	234	3 withdraw	113
Number of papers	**1110 papers**			
Remove by title	172 withdraw	62	86 withdraw	27
Number of papers	**207 papers**			
Remove by abstract	40 withdraw	22	18 withdraw	9
Number of papers	**56 papers**			
Papers not found	0 withdraw	22	0 withdraw	9
Remove by quality criteria	7 withdraw	15	0 withdraw	9
Remove by full paper	11 withdraw	4	4 withdraw	5
Extracted Papers	**9 papers**			

Q1: What modifiers are used to create games from other games?

Increased accessibility to personal computers and the expansion of the Internet, which is disseminating an increasing quantity of content, are closely related to the rise of the mod trend [23]. The community and academy are increasingly generating game adaptations, which help game developers in a variety of ways, such as recruiting new players, prolonging the life of a game, providing new perspectives for the game, and fixing bugs. In general, modifications are referred to as mods and can be viewed as alterations to the original game [6]. In general, a mod is an original game that has had one or N alterations or modifications made to its mechanics, dynamics, rules, or some of its components [4].

Mods are as diverse as the games themselves. They vary in size and complexity and can make minor adjustments to the original game or completely alter its visual design

Table IV
TRACEABILITY MATRIX OF MUTATORS AND GAMES.

Title	Year	Q1	Q2	Q3	Q4
First Stage - White Literature					
Building the Perfect Game – An Empirical Study of Game Modifications [2]	2020	X	X	X	X
To mod or not to mod—an empirical study on game modding as customer value co-creation [4]	2020		X	X	
Modding tabletop games for education [24]	2019	X	X	X	X
Migrating Java-based apo-games into a composition-based software product line [25]	2019		X	X	X
Product line architecture recovery with outlier filtering in software families: the Apo-Games case study [26]	2019	X			X
Apo-games-a case study for reverse engineering variability from cloned Java variants [27]	2018			X	X
Multi-objective optimization for reverse engineering of apo-games feature models [28]	2018		X	X	X
Visual and computational modelling of minority games [29]	2017	X	X		X
Placing value on community co-creations: A study of a video game 'modding' community [30]	2017		X	X	X
Analysis of popularity of game mods: A case study [31]	2016		X	X	
Serious mods: A case for modding in serious games pedagogy [17]	2016	X		X	
Design of a math learning game using a Minecraft mod [32]	2014	X	X	X	X
Applying exception handling patterns for user interface customization in software games modification [33]	2013	X	X	X	X
An environment to support collaborative learning by modding [34]	2014		X	X	X
Reporting about the Mod software process [35]	2012		X		X
A Role-Playing Game for a Software Engineering Lab: Developing a Product Line [35]	2012		X		X
Remix and play: Lessons from rul ts in texas hold'em and halo 2 [36]	2012		X	X	X
Modding as part of game culture [5]	2012	X		X	X
Utilizing a 3D game engine to develop a virtual design review system [37]	2011	X		X	X
Modding as an open source approach to extending computer game systems [7]	2011	X		X	X
When the game is not enough: Motivations and practices among computer game modding culture [38]	2010	X	X	X	
Modding as a basis for developing game systems [6]	2011	X		X	X
Of mods and modders: Chasing down the value of fan-based digital game modifications [39]	2009	X		X	X
Am I Mod or Not? - an Analysis of First Person Shooter Modification Culture [40]	2005	X	X	X	X
Second Stage - Gray Literature					
Unofficial patch [41]	2021	X		X	X
Mod (video gaming) [42]	2021	X			X
Appropriation & Motivation in Game Modification [43]	2020	X	X		X
Video game conversion [44]	2020	X			
Players as Content Creators the Benefits of Game Modding According to Polish Users. [45]	2019	X	X		X
Mod (video games) [46]	2018	X	X	X	X
Understanding Game Modding through Phases of Mod Development [47]	2015	X	X	X	
Does game modding require programming? [48]	2015	X			X
Computer game modders' motivations and sense of community: A mixed-methods approach [49]	2014	X		X	
Game Mods: Design, Theory and Criticism [50]	2013	X	X		
Computer game mods, modders, modding, and the mod scene [51]	2010	X	X	X	X
On modder labour, commodification of play, and mod competitions [48]	2007	X	X	X	X

[40]. Modding is the process and technique of modifying or adapting video games. It is frequently a "Do It Yourself" (DIY) strategy that teaches social and technical skills affiliated with innovation by reusing the concept of an existing game, which can be compared to opportunistic Software Reuse [27, 12]. Numerous aspects of the game, including the user interface, game items, bug fixes, characters, and regulations, are modifiable [2]. By altering the rules of a game, for instance, players are able to construct a unique gaming experience [36, 52].

Developing mods is possible by applying mutators to a game. A mutator is a modification to an existing game; for instance, applying mutator M to game G results in the creation of a new game named G [M] [53]. Depending on the number of mutators utilized, a game may be classified in a variety of ways. There are numerous adaptations and modifications, each serving a distinct purpose [2, 6]. Each of them will be described in more detail in the following [1-3][8][16-27].

1) **Interface customization:** The interfaces are designed to emphasize the visual component of the game in order to enhance the experience. This customization entails making changes to the visual element, such as remodeling the accessories, skin, shader, or animation of a character or a game map, altering the game's colors, or altering the information displayed on the screen.

2) **Partial Conversions:** Add a new map, a new character, and a new item; increase the game's pace; add small mechanisms, bots, and rules. It is still possible to classify partial modifiers according to the modifications they execute.
    a) **Mutators/tweaks:** Modify or add restricted features that have no effect on the game's functionality or mechanics. They may include modifying the game's theme song, increasing the game's speed, or modifying some graphic elements and minor rules.
    b) **Add-ons:** They serve as supplementary elements within the game, performing minor adjustments such as modifying the theme's music, accelerating the game, or adjusting minor graphical components and rules.
    c) **Mods:** They are the intersection of the previous two, as they retain the capacity to change rules and configurations.
3) **Total Conversions:** Certain changes are so drastic that they result in the creation of new games. A well-known conversion is the CounterStrike mod, which was based on Half-Life. In general, the number of modifiers used differentiates a partial conversion from a complete conversion. When a significant number of modifiers are applied to the point where something new is generated, a complete conversion occurs.
4) **Machinima:** It could be seen as the outcome of changes that influence the visual replay of game usage sessions. In this type of modification, games are used for other purposes, such as telling a story, making a movie, or replicating a gaming experience.
5) **Patch:** They frequently concentrate on addressing unresolved problems and creating technical enhancements. This modification is known as an unofficial or fan patch when it is created by a community.

Q2: What characteristics are needed to derive a game?

A game is a type of software development in which designers, developers, and software engineers work together to create an experience for players to live through the game [30]. Once the game is out, the contributors devote their time to updating and adding content to the main game. Modifications may include new game models, textures, music, and mechanisms, as well as complete remakes [30].

There are two primary methodologies for mod development. The first scenario occurs when there is a need for expansion in a particular game by introducing new elements, while the second scenario occurs while seeking games that offer similar characteristics to those wanted in the game under development [24]. Both need the same characteristics.

A game is made up of components that work together to generate the final output. The required qualities for their construction can be determined by defining games. Games are activities that occur in an abstract environment where decisions, actions, and rules are developed with the objective of accomplishing a leisure activity in the form of entertainment or amusement [54]. On this premise, the following aspects must be decided prior to the construction of any game: rules, actions, behaviors, objective, game loop, difficulty, and rewards [24, 9].

Each of the characteristics necessary for the interpretation and evolution of a game will be exemplified below. These features were divided into four broad categories that capture the attributes of the games at a higher level of abstraction. It should be noted that game mechanics were previously divided into actions and behaviors [17-19][23][24][29][32].

1) **Avatar**
    a) **Operation rules**: Rules about the player. E.g.: the player can only carry one weapon at a time [4][14][18][28][33-35];
    b) **Transition rules/states**: Understanding the character's state transitions. E.g.: the player can only shoot if he/she has a weapon in his/her hand [24, 32, 4, 28, 35, 26];
    c) **Actions**: Commands that can be executed by the character. E.g.: shooting and walking [2, 29, 32, 35, 4, 26].
1) **Game world**
    a) **Levels**: The game's stages. Strongly influenced by the gameplay that can change from one stage to the next [24, 28, 32, 37, 35, 4, 26];
    b) **Rules of objects**: Rules of the objects contained in the world. E.g.: when an object must be locked or unlocked [2, 24, 29, 55, 4, 26];
    c) **Behavioral rules**: Rules of behavior that the world can exhibit. Eg .: if the player collects a specific item it can start to rain [24, 29, 37, 56, 30, 26];
    d) **Temporal states**: It works like a state machine; depending on the world's state, it can only go to a specific one [2, 24, 37, 56, 26];
    e) **Mission**: What you want to achieve/complete [2, 24, 32, 37, 30, 4, 26];
    f) **Obstacles**: What you must overcome in the game, its difficulties [37, 17, 35, 56, 30, 4, 26].
1) **Game play**
    a) **Winning and losing conditions**: Conditions to win or lose the game [24, 17, 56, 40, 4, 26];
    b) **Strategic dilemmas**: Strategies that can be used in the game. E.g.: combo attacks [24, 17, 56, 40, 4, 26];
    c) **Chains of actions**: Chain of actions that can be combined. Eg.: player action with a map action [24, 35, 56, 4].
1) **General features**
    a) **Rules**: Encapsulates the logic inside the system [2, 24, 28, 29, 34, 36, 4, 26];
    b) **Score**: The points obtained by the player throughout the game [24, 28, 35, 56, 4, 26];

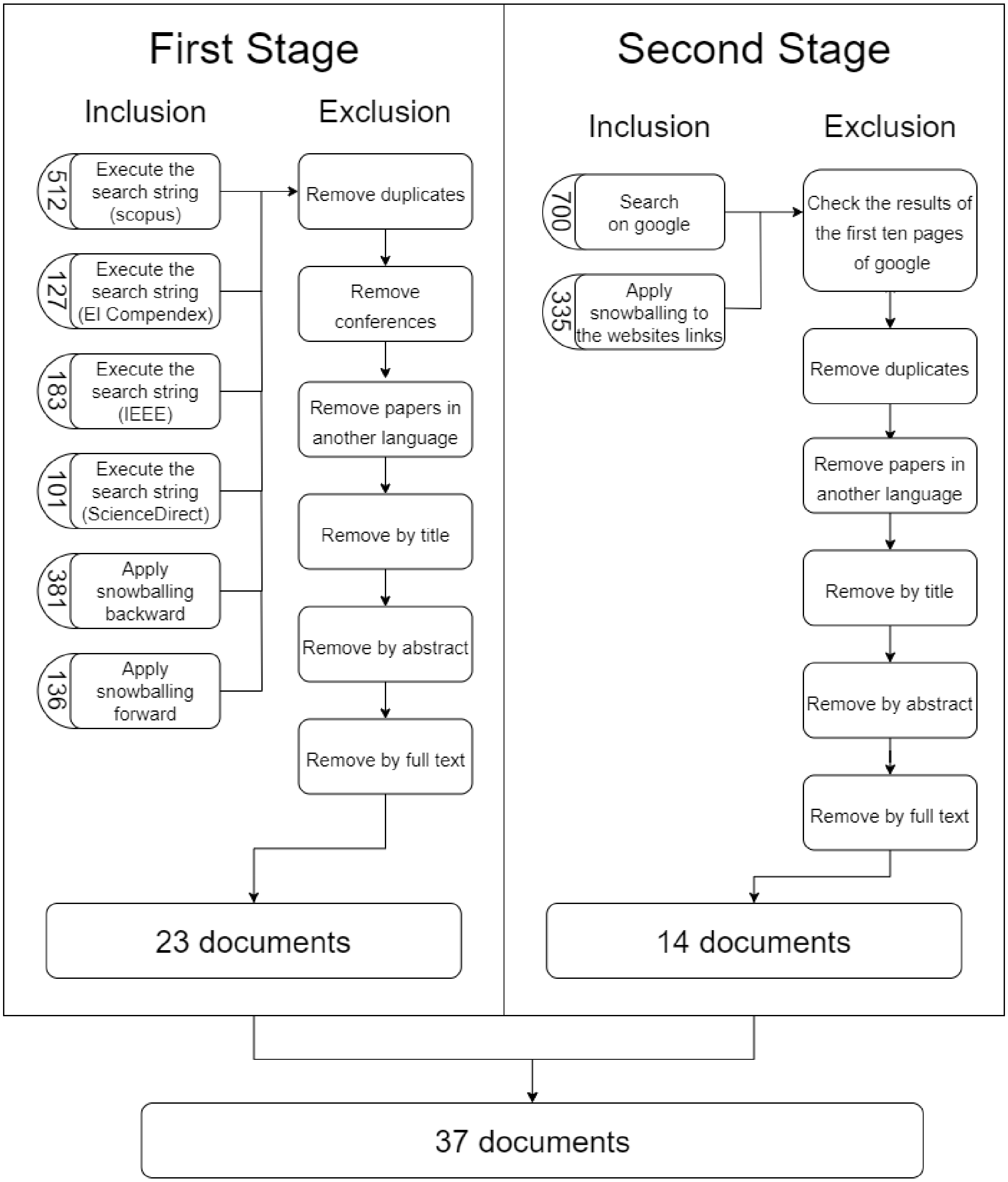

First Stage
Inclusion
Exclusion
512 Execute the search string (scopus)
127 Execute the search string (EI Compendex)
183 Execute the search string (IEEE)
101 Execute the search string (ScienceDirect)
381 Apply snowballing backward
136 Apply snowballing forward
Remove duplicates
Remove conferences
Remove papers in another language
Remove by title
Remove by abstract
Remove by full text
23 documents
Second Stage
Inclusion
Exclusion
700 Search on google
335 Apply snowballing to the websites links
Check the results of the first ten pages of google
Remove duplicates
Remove papers in another language
Remove by title
Remove by abstract
Remove by full text
14 documents
37 documents

Figure 1. Search flow of research on mutators and games.

c) **Behaviors**: Commands that are executed by the system [2, 24, 28, 56, 4, 26];
d) **Goal**: What you want to achieve/complete [2, 24, 32, 37, 4, 26];
e) **Challenge**: What must be accomplished to achieve the goal [2, 24, 28, 17, 4, 26];
f) **Rewards**: reaching the goal [2, 24, 29, 34, 4, 26];
g) **Game loop**: Flow of engagement of the game. It is the execution of the game where the player seeks a goal by executing a challenge and being rewarded with something [2, 28, 35, 56, 4];
h) **Interface**: The visual of the game, the game's sprites, and graphics [28, 29, 32, 34, 37, 17, 26];
i) **Entities**: Objects and elements instantiated within the game [28, 29, 56, 40, 26].

Q3: What are the advantages and difficulties of creating games from others?

Generalizing mod developers' intentions is difficult. There are several elements that contribute to a user producing a mod.

Attempting new things, resolving bugs, creating new characters, increasing the difficulty of the game, gaining advantages in the game, extending the game's life cycle, the software was originally designed for a significantly different environment and may require improvement, the official developer is unable to deal with the problems, and so on are among the most significant ones [2, 47].

Modifiers, like games, are complex and time-consuming to create. The time it takes to create a mod varies greatly. The construction process might vary in duration, ranging from a few days to a somewhat longer period, while offering the advantage of using reusable components. As previously mentioned, creating a game can be incredibly time consuming and can take years. However, the time necessary to release a mod is far shorter [2]. Mods allow the community to add to the original game. Depending on the nature of the mod, it may only require one or several releases. For example, a mod that improves the texture of a game may only require one version.

The potential to increase the longevity of games is another advantage that can be ascribed to the employment of modifiers. Every game has an effective life cycle. Modifiers, on the other hand, can extend the life of the game by adding additional instructions, characters, levels, and other factors, giving players more areas to explore [2, 4, 38]. Using the same logic, modifications may help boost sales, income, and profits for original games, as many people purchase the original game in order to play the mod [2, 52, 47].

Another significant advantage of modifications is their ability to draw new players to the game, so extending its longevity. For example, Dota 2 was a Warcraft mod that reached 450,000 daily players five years after its debut and 16 years after the original game's release. As a result, the game's player base and longevity grow [2, 39].

Along with the benefits described above, several additional are still directly tied to the community member who worked on the modification. Among the primary ones that stand out are the user's expression, communication between the firm and the end-user, and diversity of the game via end-user ideas [4]. Mods have become such a common practice in the community that they may now be considered a form of culture, allowing the user to incorporate their experiences into the game [5, 7].

When a player buys a game, he or she obtains a license to use the product. This license is structured in the manner of a copyright-based agreement [51]. Mods made by the community are susceptible to contract violations, placing their authors at considerable danger of being held liable for their actions, which frequently involve scamming games, exploiting product faults, or committing copyright infringement [6].

Last but not least, this mod technique has gotten so widespread that many large game production firms have opted to enlist members of the gaming community in order to cut development costs and risks. This cooperation enables the players to pool their unique perspectives and extensive experience and skills, therefore boosting the quality of the innovation without incurring additional resources, as well as certifying that it was what the community expected in terms of the game [30, 33].

Table V
ADVANTAGES AND DISADVANTAGES OF USING MODS.

Advantages / Disadvantages	Papers
Communication between the company and the end-user	[2, 41, 47, 49]
Diversification of the game through end-user ideas	[2, 45, 39]
Decreases the risk of game bugs	[47, 4, 41, 51, 38, 33, 49, 2]
Create new instructions, characters, levels, and other elements, providing players with new aspects to explore	[4, 5, 7]
Increase the game life cycle	[2, 4, 38, 51, 47, 33]
Shorter development time	[30, 33, 4]
Lower development cost	[51, 47, 33, 49, 41]
Increase the number of players of the original game	[2, 47, 49]
High initial investment	[27, 28]

Despite all the advantages described so far, some difficulties and challenges must be observed when creating mods. The first and main problem is the initial investment to produce a mod, which is necessary to understand the source code, reverse engineering and extract its features [28, 27]. Following this line of reasoning, some studies have already been carried out using the product line. However, this approach also requires an initial investment to conceptualize the initial features of the project [28, 27].

Table V demonstrates the advantages and disadvantages of using mods. The green color shows the advantages and the red the disadvantages.

Q4: What tools or frameworks support these changes?

Numerous frameworks and tools facilitate the building of modifications. However, the most prevalent technique of mod development so far has been cloning and do-it-yourself. The modder selects the basic game to be updated, verifies the characteristics he/she wants to modify, and then produces the new game [27]. This less complex strategy is referred to opportunistic reuse or ad-hoc reuse, and it comprises cloning, copying, and straining. Opportunistic reuse provides immediate advantages and produces the desired outcome. However, the quality of the project is not a priority, significant reworking leads to unexpected behavior and an unstable software structure [26].

Typically, games are changed using tools that enable access to an unencrypted internal representation of the game program. While it may appear as though game developers would aim to discourage consumers from customizing their games, this is not the case. Developers of video games are increasingly providing software tools for customizing their products in order to boost sales and market share [47]. Software development kits (SDKs) for games/domains supplied to users by game development studios represent a modern business approach for engaging users and assisting in product innovation outside the studio [6, 45, 30]. In addition to SDKs, which are the most common way of accessing the game's source code, several other platforms provide access to the game's source code and allow modifications. Among the main ones are the Creation Kit, GECK, Construction Set, MODKit, REDKit, Modbuddy, and D'jinni [2].

Another possibility for the development of modifiers is through free software games, in which the end user has complete access to the game's source code and may modify it as desired [6]. However, this strategy is used by small businesses or anonymous developers.

There are firms that assist and encourage the production of modifications with the goal of reducing problems, improving the game's quality and consistency, and generating new ideas. This technique leverages the users' ideas and wants to generate improvements for the game sold. The Unreal engine was created to provide access to all of its technology's components. This enabled it to host multiple events dubbed Unreal Tournaments, in which the developer may express his/her creativity while developing his/her mods [40, 2]. Other companies permit the construction of modifications as well, although without providing direct access to the components. For instance, Blizzard Entertainment's World of Warcraft has a UI modification tool that enables add-ons to modify the user interface panel, resulting in an enhanced gameplay experience. But these add-ons do not modify or convert the game into something entirely different since Blizzard seeks to ensure that players have access to the same configuration and mechanics as the original game [51].

In adiction, there are developers that produce mods by reverse engineering the source code of the original game. It is worth noting that this method for developing modifications is unlawful and violates the copyright of the original games [41].

Finally, single mod distribution platform may include several modifications for a single game. They all, however, adapt the same basic game. These platforms must be demonstrated in some way, such as demonstrating which files were changed in each mod, or if one mod is compatible with another, as both can change the same original game file [2].

Due to the large number of game variations generated based on an original game, maintenance can become difficult, and businesses may consider transitioning to a line of software products, referred to as an extractive method, to assist with mass game production [27, 35, 25, 28, 26].

Finally, there are several sites that support mods, integrating the community and providing thousands of games. Among the main games are gamemodding (http://www.gamemodding.net), moddb (https://www.moddb.com), modsonline (http://modsonline.com/), among others [2, 56].

V. DISCUSSION

The research revealed that the community has been utilizing methods of reuse to create games. Modding, the practice of game reuse development using existing games as a base, is widely utilized by both the gaming community and companies to boost their games or expand their game collection. Mods are a common means of self-expression in the community, however, they are typically created in a casually manner and are frequently linked to opportunistic reuse, frequently practiced out through clone-and-own techniques. Search results reveal other techniques for creating games with SR, like software componentization, enabling integration with popular game engines. Nevertheless, techniques like SPL and MDD are currently in the testing phases, with limited instruments for use.

Utilizing reuse in software development can provide several advantages, such as decreasing development time, enhancing software quality, and facilitating quicker and easier product expansion. Considering these remarkable benefits, it is feasible to relate them to the challenges in game development, such as long development time, risk of numerous defects, and complexity of generating new versions of the original game.

Various methods exist for implementing software reuse, ranging from clone-and-own to product lines. Each of these methods has distinct benefits and suits certain goals. Each of these topics will now be illustrated with a focus on developing games. All the following observations were made with consideration for the development of mods and game expansion.

- **Clone-and-own:**
  - **Context of use:** This method could be used for modest projects if expanding the game is not part of the initial plan.
  - **Advantage:** This approach provides an excellent initial return by reusing a significant portion of the existing game.
  - **Disadvantage:** Future maintenance and upgrades can generate significant expenses if the same changes and improvements are applied to multiple projects simultaneously.
  - **Example:** adapt particular elements of an open-source game.
- **Componentization:**
  - **Context of use:** This method is suitable for tasks of medium to big complexity. Expansion of the original game is premeditated, but new games can be created by reusing its components. The main objective is to create new games by utilizing its components again.
  - **Advantage:** this strategy is widespread and utilized by numerous engines. Any alteration to a component will result in all games utilizing that component being updated.
  - **Disadvantage:** does not allow expansion of games directly.
  - **Example:** in a First-Person Shooter (FPS) game, all moves and a lot of actions may be broken down into smaller parts because they are the same in all games.
- **MDD:**
  - **Context of use:** This method is appropriate for medium to large projects. The original game's expansion was not preplanned. This strategy is popular because it begins implementation only after a well-

developed model has been generated through numerous iterations.
  - **Advantage:** platforms based on this production approach enable anyone with basic programming knowledge to produce full-length games.
  - **Disadvantage:** unlike the others, it is not a development pattern, but a development process.
  - **Example:** use in projects with poorly specified criteria and a high probability of modification due to the ability to change the models before beginning the project.
- **SPL:**
  - **Context of use:** this method is appropriate for medium to large projects. The expansion of the original game was planned before.
  - **Advantage:** several games can be derived from a single game by selecting game features.
  - **Disadvantage:** few tools and platforms available.
  - **Example:** the product line makes it easy to make games that can be played on a variety of platforms. Think of a game that needs both a mobile and a desktop form.

VI. CONCLUSION

Game companies are growing in size, generating billions of dollars each year, releasing many titles each year, and attracting fans of all ages and genres. However, as has been seen so far, developing a game may be a long process that might take years to complete. However, the gaming community is rising daily. With such a vast user community, some members may experience anxiety or dissatisfaction at the prospect of having to wait so long for a game to be published.

With a little online search, it is possible to locate multiple games for sale and several websites that provide modifications for them. As previously mentioned, a mod may be defined as a modification made to a game and depending on the level of this modification this mod can receive different names, such as: patches, tweaks, add-ons, among others. This method of modifying games can result in a number of benefits for the company that generated the original games. The benefits are numerous, and some businesses even encourage this practice. Among the primary benefits are an increase in users, an increase in sales, and an increase in the game's longevity, among others. However, the study revealed that the process of developing a mod might be expensive and ad-hoc.

It is noted the presence of tools and frameworks that support modifications, ranging from clones to SDKs and tools made accessible by the game's developer, among other techniques. However, these tools are frequently associated with a number of difficulties, including a large initial investment, a steep learning curve, the requirement to comprehend the source code, and the fact that the majority of these techniques are limited to the creation of basic games.

Throughout the investigation, multiple papers were found that contrasted ad-hoc mod development with opportunistic reuse, in which software is built using similar strategies of small modifications and copying and pasting [57, 58]. Demonstrating once again the need of systematizing the process of mod development. Additionally, it was seen in these same papers an early application of Software Reuse (SR) for game building. However, it was employed superficially and exploratorily.

As is already known, SR can bring several types of advantages in the construction of software in general, from systematization of development to an increase in delivery speed and cost reduction. It was observed that these advantages were being highlighted in some articles through the research, but, however, the approaches demonstrated were of an exploratory nature, with the exception of software componentization, an approach that has already been used by programmers and is even found in some more current advanced engines [59, 60].

Initial discussions about the possible use of SR approaches in game development were made in the discussion section, however, it was decided to conduct another research that would complement this study. The proposed study sought to determine which SR techniques are being utilized to create games or mods.

REFERENCES

[1] D. Castro and C. Werner, "A multivocal review on derivation games," *The 2023 IARIA Annual Congress on Frontiers in Science, Technology, Services, and Applications, IARIA Congress*, pp. 144–149, November 2023.

[2] D. Lee, D. Lin, C.-P. Bezemer, and A. E. Hassan, "Building the perfect game–an empirical study of game modifications," *Empirical Software Engineering*, pp. 1–34, 2020.

[3] S. Pashkov, "Video game industry market analysis: Approaches that resulted in industry success and high demand," 2021.

[4] K. Bilińska, A. Dewalska-Opitek, and M. Hofman-Kohlmeyer, "To mod or not to mod—an empirical study on game modding as customer value co-creation," *Sustainability*, vol. 12, no. 21, p. 9014, 2020.

[5] A. Unger, "Modding as part of game culture," *Computer Games and New Media Cultures*, pp. 509–523, 2012.

[6] W. Scacchi, "Modding as a basis for developing game systems," *Proceedings of the 1st international workshop on Games and software engineering*, pp. 5–8, 2011.

[7] ——, "Modding as an open source approach to extending computer game systems," *Proceedings of IFIP International Conference on Open Source Systems*, pp. 62–74, 2011.

[8] V. Garousi, M. Felderer, and M. V. Mäntylä, "Guidelines for including grey literature and conducting multivocal literature reviews in software engineering," *Information and Software Technology*, vol. 106, pp. 101–121, 2019.

[9] J. Ehrmann, C. Lewis, and P. Lewis, "Homo ludens revisited," *Yale French Studies*, no. 41, pp. 31–57, 1968.

[10] R. Hunicke, M. LeBlanc, and R. Zubek, "Mda: A formal approach to game design and game research," in *Proceed-*

ings of the AAAI Workshop on Challenges in Game AI*, vol. 4, no. 1, 2004.

[11] B. Randell, "Software engineering in 1968," in *Proceedings of the 4th International Conference on Software Engineering*, ser. ICSE'79. IEEE Press, 1979, p. 1–10.

[12] C. W. Krueger, "Software reuse," *ACM Computing Surveys (CSUR)*, vol. 24, no. 2, pp. 131–183, 1992.

[13] J. Sametinger, *Software Engineering with Reusable Components*. Springer Science & Business Media, 1997.

[14] N. Maia, A. P. T. Bacelo, and C. M. L. Werner, "Odysseymda: A transformational approach to component models." *Proceedings of Internacional Conference on Software Engineering and Knowledge Engineering*, pp. 9–14, 2007.

[15] B. Kitchenham, O. P. Brereton, D. Budgen, M. Turner, J. Bailey, and S. Linkman, "Systematic literature reviews in software engineering–a systematic literature review," *Information and software technology*, vol. 51, no. 1, pp. 7–15, 2009.

[16] M. Petticrew and H. Roberts, *Systematic reviews in the social sciences: A practical guide*. John Wiley & Sons, 2008.

[17] V. McArthur and R. J. Teather, "Serious mods: A case for modding in serious games pedagogy," *Proceedings of IEEE Games Entertainment Media Conference (GEM)*, pp. 1–4, 2015.

[18] T. M. Moraes and A. d. SOUZA, "Revisão sistemática sobre a comunicação dentro do processo de desenvolvimento de software," *Universidade Federal de Goiás-GO*, p. 57, 2011.

[19] R. C. Motta, K. M. de Oliveira, and G. H. Travassos, "Characterizing interoperability in context-aware software systems," *Proceedings of VI Brazilian Symposium on Computing Systems Engineering (SBESC)*, pp. 203–208, 2016.

[20] S. Matalonga, F. Rodrigues, and G. H. Travassos, "Characterizing testing methods for context-aware software systems: Results from a quasi-systematic literature review," *Journal of Systems and Software*, vol. 131, pp. 1–21, 2017.

[21] C. Wohlin, "Guidelines for snowballing in systematic literature studies and a replication in software engineering," in *Proceedings of the 18th International Conference on Evaluation and Assessment in Software Engineering*, 2014, pp. 1–10.

[22] Y. Lincoln and E. Guba, "Naturalistic inquiry. encyclopedia of research design. 2455 teller road, thousand oaks california 91320 united states," 2016.

[23] O. Sotamaa, "On modder labour, commodification of play, and mod competitions," *First Monday*, vol. 12, no. 9, 2007.

[24] D. Abbott, "Modding tabletop games for education," *Proceedings of International Conference on Games and Learning Alliance*, pp. 318–329, 2018.

[25] J. Debbiche, O. Lignell, J. Krüger, and T. Berger, "Migrating java-based apo-games into a composition-based software product line," in *Proceedings of the 23rd International Systems and Software Product LineConference, vol. A*, 2019, pp. 98–102.

[26] C. Lima, W. K. Assunção, J. Martinez, W. Mendonça, I. C. Machado, and C. F. Chavez, "Product line architecture recovery with outlier filtering in software families: the apo-games case study," *Journal of the Brazilian Computer Society*, vol. 25, no. 1, pp. 1–17, 2019.

[27] J. Krüger, W. Fenske, T. Thüm, D. Aporius, G. Saake, and T. Leich, "Apo-games: a case study for reverse engineering variability from cloned java variants," in *Proceedings of the 22nd International Systems and Software Product LineConference, vol. 1*, 2018, pp. 251–256.

[28] W. D. Mendonça, W. K. Assunção, and L. Linsbauer, "Multi-objective optimization for reverse engineering of apo-games feature models," in *Proceedings of the 22nd International Systems and Software Product LineConference, vol. 1*, 2018, pp. 279–283.

[29] R. Damaševičius and D. Ašeriškis, "Visual and computational modelling of minority games," *TEM J*, vol. 6, no. 1, pp. 108–116, 2017.

[30] L. Poretski and O. Arazy, "Placing value on community co-creations: A study of a video game'modding'community," in *Proceedings of the 2017 ACM Conference on Computer Supported Cooperative Work and Social Computing*, 2017, pp. 480–491.

[31] T. Dey, J. L. Massengill, and A. Mockus, "Analysis of popularity of game mods: A case study," in *Proceedings of the 2016 Annual Symposium on Computer-Human Interaction in Play Companion Extended Abstracts*, 2016, pp. 133–139.

[32] R. Al-Washmi, J. Bana, I. Knight, E. Benson, O. A. A. Kerr, P. Blanchfield, and G. Hopkins, "Design of a math learning game using a minecraft mod," in *European Conference on Games Based Learning*, vol. 1. Academic Conferences International Limited, 2014, p. 10.

[33] T. Tengtrirat and N. Prompoon, "Applying exception handling patterns for user interface customization in software games modification," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 1, 2013.

[34] S. George, É. Lavoué, and B. Monterrat, "An environment to support collaborative learning by modding," in *European Conference on Technology Enhanced Learning*. Springer, 2013, pp. 111–124.

[35] S. Zuppiroli, P. Ciancarini, and M. Gabbrielli, "A role-playing game for a software engineering lab: Developing a product line," in *2012 IEEE 25th Conference on Software Engineering Education and Training*. IEEE, 2012, pp. 13–22.

[36] G. Cheung and J. Huang, "Remix and play: lessons from rule variants in texas hold'em and halo 2," in *Proceedings of the ACM 2012 Conference on Computer Supported Cooperative Work*, 2012, pp. 559–568.

[37] M. F. Shiratuddin and W. Thabet, "Utilizing a 3d game engine to develop a virtual design review system," 2011.

[38] O. Sotamaa, "When the game is not enough: Motivations and practices among computer game modding culture," *Games and Culture*, vol. 5, no. 3, pp. 239–255, 2010.

[39] H. Postigo, "Of mods and modders: Chasing down the value of fan-based digital game modifications," *Games and Culture*, vol. 2, no. 4, pp. 300–313, 2007.

[40] D. B. Nieborg, "Am I mod or not?—An analysis of first person shooter modification culture," in *Creative Gamers Seminar—Exploring Participatory Culture in Gaming, University of Tampere, Finland (14–15 January)*, 2005.

[41] J. Kolbert, "Unofficial patch," https://en.wikipedia.org/wiki/Unofficial_patch, online; accessed 09 March 2021.

[42] R. F. Patterson, "Mod (video gaming)," https://civilization.fandom.com/wiki/Mod_(video_gaming), online; accessed 05 March 2021.

[43] C. Weeke, "Appropriation & motivation in game modification," *Erasmus University Thesis Repository*, 2020.

[44] M. Magioladitis, "Video game conversion," https://en.wikipedia.org/wiki/Video_game_conversion, online; accessed 07 March 2021.

[45] M. M. Hofman-Kohlmeyer, "Players as content creators. the benefits of game modding according to polish users," *International Scientific Journal News*, vol. 2, pp. 8–26, 2019.

[46] N. Cleland, "Mod (video games)," https://en.wikipedia.org/wiki/Mod_(video_games), online; accessed 04 March 2021.

[47] S. Agarwal and P. Seetharaman, "Understanding game modding through phases of mod development." *Proceedings of ICEIS*, pp. 114–121, 2015.

[48] R. Ramadan, "Does game modding require programming?" https://www.quora.com/Does-game-modding-require-programming, online; accessed 09 March 2021.

[49] N. Poor, "Computer game modders' motivations and sense of community: A mixed-methods approach," *New media & society*, vol. 16, no. 8, pp. 1249–1267, 2014.

[50] E. Champion, *Game mods: design, theory and criticism*, Lulu.com, 2013.

[51] S. Walt, "Computer game mods, modders, modding, and the mod scene," *First Monda*, vol. 15, no. 5, 2010.

[52] W. Scacchi, "Computer game mods, modders, modding, and the mod scene," *First Monday*, 2010.

[53] B. T. Claude Chaunier, Torben Mogensen, "Mutators," http://www.di.fc.ul.pt/~jpn/cv/mutators.htm, online; accessed 04 March 2021.

[54] G. Xexéo, A. Carmo, A. Acioli, B. Taucei, C. DIpolitto, E. Mangeli, J. Kritz, L. Costa, and R. Monclar, "What are games?" *LUDES. Rio de Janeiro*, vol. 1, pp. 1–30, 2013. (In Portuguese).

[55] J. Åkesson, S. Nilsson, J. Krüger, and T. Berger, "Migrating the android apo-games into an annotation-based software product line," in *Proceedings of the 23rd International Systems and Software Product LineConference, vol. A*, 2019, pp. 103–107.

[56] G. A. Cignoni, "Reporting about the mod software process," in *European Workshop on Software Process Technology*. Springer, 2001, pp. 242–245.

[57] W. Fenske, J. Meinicke, S. Schulze, S. Schulze, and G. Saake, "Variant-preserving refactorings for migrating cloned products to a product line," in *2017 IEEE 24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*. IEEE, 2017, pp. 316–326.

[58] M. Sierra, M. C. Pabón, L. Rincón, A. Navarro-Newball, and D. Linares, "A comparative analysis of game engines to develop core assets for a software product line of mini-games," in *International Conference on Software and Systems Reuse*. Springer, 2019, pp. 64–74.

[59] N. Aouadi, P. Pernelle, C. B. Amar, T. Carron, and S. Talbot, "Models and mechanisms for implementing playful scenarios," in *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*. IEEE, 2016, pp. 1–8.

[60] E. Albassam and H. Gomaa, "Applying software product lines to multiplatform video games," in *2013 3rd International Workshop on Games and Software Engineering: Engineering Computer Games to Enable Positive, Progressive Change (GAS)*. IEEE, 2013, pp. 1–7.

Model-Supported Software Creation: Extending Model-Driven Software Engineering with Non-Formal Artifacts and Transformations

Hans-Werner Sehring
Department of Computer Science
Nordakademie
Elmshorn, Germany
e-mail: sehring@nordakademie.de

Abstract—**Software typically is developed based on descriptions of a relevant section of the real world, the problem at hand as well as the software to be built for its solution. Methodologies and tools have evolved to create and manage such descriptions, and to finally implement software as specified. Model-Driven Software Engineering (MDSE) is one approach of model management. A series of models that build upon each other by means of model transformation is used to describe a software solution in increasing detail. While the application domain and the software solution under consideration are reflected by such models, other aspects of a software project are not always considered on equal level. Examples are the business side of the project that usually exceeds the software creation part, creative activities like user interface design, and aspects of the operation of the software. In this article, we discuss aspects of extending MDSE towards a holistic approach that includes additional phases of software engineering and the incorporation of models that are either defined in specific notations used by experts or that do not allow formalized model transformations. The approach relies on artifacts that are created using a heterogeneous set of languages. These artifacts are described by formal models that add semantics and that relate the informal artifacts. For such an approach, we coin the term "model-supported software creation" in this article.**

Keywords—model-driven software engineering; model-driven architecture; software engineering; software architecture

I. INTRODUCTION

Software is, in most of the cases, used to represent and solve real-world problems. In order to be able to do so, a relevant section of the real world needs to be captured, and the problem as well as its solution need to be described in sufficient detail. This includes defined requirements, test cases, conceptual models, domain models, etc.

Methodologies and tools have evolved that capture problems and solutions, model the real world with respect to the problem at hand, and finally allow implementing software with respect to such a model.

Classical software engineering has a typical sequence of an analysis phase, resulting in requirements, design phases, resulting in solution designs, and implementation phases, resulting in working software. In agile approaches, these phases may be very condensed. The artifacts (descriptions, models, code, etc.) created in each phase build upon each other. Still, they are formally unrelated. Those artifacts contributing to a phase consider the artifacts from previous phases, though.

The various description artifacts involved in software engineering processes call for means to manage these descriptions. In particular, they have to be related to each other to reach goals like, for example, those of coherence and traceability.

Model-Driven Software Engineering (*MDSE*) or *Model-Driven Software Development* (*MDSD*) is one approach to a more formal management of artifacts. A series of models that build upon each other is used to describe a software solution in increasing detail. Typically, the models are refined or transformed up to the point where actual running software can be generated out of the most precise model.

Software engineering, and thus MDSE, at best captures the whole software lifecycle. Ultimately, all development steps are captured by a *holistic* MDSE approach. In this article, we study two dimensions in which to extend typical MDSE approaches. We use the name *Model-Supported Software Creation* (*MSSC*) for an accordingly extended kind of MDSE. This article is an extension of the presentation of the first ideas towards holistic software (project) models in [1].

While MDSE gained a fair amount of attention, it is not equally successful in all application domains [2]. We see two main obstacles to applying MDSE in some areas: the heterogeneity of modeling artifacts and the stages of software development that are covered by a software engineering process.

1) MDSE is well-suited for formal domains and for computation-centric solutions. But is is not equally well applicable to software development processes with a high degree of creativity involved. For example, it is feasible to model technical domains that are based on mathematics and physics. But it is less practical to formally model solutions with a focus on creative and subjective aspects. Human-machine interaction (online shops, for example) or content-centric applications (personalized marketing websites, for example) are examples found in typical customer-facing commerce systems.
2) MDSE focuses on the stages of a project where the actual software is specified and implemented. Most approaches start with defined requirements. Projects include more tasks than just software creation alone. There are earlier stages in which (business) goals are set and a decision is made to start a project, and there are stages that follow software implementation, like operations and maintenance.

In order to incorporate these aspects in MDSE, we study a modeling approach that allows incorporating models in varying notations and modeling approaches that do fully rely on formalized model transformations. It is based on models that are created using a heterogeneous set of languages and that are used to add semantics to and that relate informal artifacts.

Section II of this paper revisits some approaches to MDSE. Additional demand for modeling that exceeds software modeling is studied in Section III (extended demand for additional project phases) and in Section IV (requirements to the integration of both formal and informal models). Section V presents the *Minimalistic Meta Modeling Language* (*M³L*) that we use as the basis for first experiments with holistic MSSC models presented in Section VI. We conclude the paper in Section VII.

II. MODEL-DRIVEN SOFTWARE ENGINEERING

Various approaches to software generation from models are discussed. In this section, we briefly revisit some of these.

A. Model-Driven Architecture

The *Model-Driven Architecture* (*MDA*) [3] of the Object Management Group (OMG) is an early and well received proposal for an MDSE approach. It assumes models to be created on (originally) three levels of abstraction. A *Computation-Independent Model* (*CIM*; this term is not used in current specifications) describes the software to be developed from the perspective of the subject domain, as domain concepts or requirements. It typically is an informal description, for example, done in natural language. A first formal model is a *Platform-Independent Model* (*PIM*), formulated in the MDA's *Meta Object Facility* (*MOF*). It is transformed into a *Platform-Specific Model* (*PSM*) that in turn is used to generate a working implementation. Model transformations are specified using *Query View Transform* (*QVT*) based on MOF.

B. Software Generation

Software generation has gained particular attention since this step in an MDSE process can well be formalized.

a) Metaprogramming: Programs that generate programs are an obvious means to software generation. The development of such generators tends to be costly, but results may be targeted optimally to the application at hand.

b) Templates: Code with repeating structures can be formulated as templates with parameters for the variations of that uniform code. For *Concept-Oriented Content Management* [4], for example, code for CRUD operations is generated. This code does not differ in functionality, but in the data types used for domain entities.

c) Generative AI: The currently emerging generative AI approaches based on large languages models provide another means to generate code from descriptions. Based on a library of samples, they allow interactively generating code from less formal descriptions, in particular natural language expressions.

C. Domain-specific Languages

Languages can be associated with metamodels [5]. This means that a model of a software application can be expressed by a language for a subject domain. Such a language is called a *Domain-Specific Language* (*DSL*).

The software generation process is simplified to defining an application using a DSL, allowing to define the application in terms of the subject domain. There is a trade-off regarding the degree of abstraction: The more domain knowledge is put into the DSL, the simpler it is to define an application. But a more specialized DSL also means that the range of application that can be defined becomes more limited.

D. Generic Software

The aim of MDSE and MSSC is custom software that is tailored to solve one specific problem. Generic software, on the other hand, encapsulates some domain knowledge that is applicable in a set of scenarios.

The concrete application is defined by setting parameters of the generic software. The application areas of generic software are defined by the degree to which domain knowledge was generalized and parameterized.

There are varying degrees of parameterization. This relates to so-called low code and no code approaches. These are also based on a generalized software that maps a section of the real world, and they allow software to be customized within the limits of the chosen section.

III. MDSE FOR THE FULL SOFTWARE LIFECYCLE

On top of software models as provided by approaches like the ones presented in the preceding section, there are additional aspects of (software) projects that have to be captured in a holistic software engineering process. In this section, we outline typical project activities and intermediate results.

Further artifacts play a role for these aspects, and they call for additional model contributions.

We call sets of project contributions that logically belong together a *modeling stage*. This term shall reflect the fact that models build up upon each other, and not in a temporal sense as terms like *project phase* would indicate.

Table I gives an overview over typical stages of software creation and some examples of artifacts they deal with.

A. Business Goals, Project Goals, and Constraints

The purpose of software typically is not just to be useful by meeting the requirements, but it contributes to some business goals. At least in commercial applications, business goals are defined upfront, and software may be one part of a solution to reach these goals.

Software modeling in the MDSE sense starts at the point where there is consensus about the kind of software to be developed. In fact, projects start at an earlier stage at which a (business) need arises. In a commercial setting, this may be, for example, increased revenue, a certain number of new customers, or some degree of customer satisfaction. A solution approach is not given. At this stage, it is not even decided that new or improved software will be part of the solution.

Therefore, on top of a software project, a business endeavor is pursued. The business goals will finally be the main criteria to measure project success. To this end, these goals need to be precisely formulated so that their impact on the software

TABLE I. STAGES OF SOFTWARE CREATION

Modeling stage	Model entities on the stage
(Business) Goals	KPIs OKRs
Subject domain model	Information architecture Interaction design Wireframes Processes, data flows
Requirements	Solution hypothesis Functional ~ Non-functional ~ Customer journeys Touch points
Solution architecture	Interfaces High-level architecture Functional mapping
Software architecture(s)	Components Communication between those components Interfaces to the environment Constraints of the resulting software system Requirements met by the architecture Rationale behind architecture decisions
Code	Metaprogramming Software generators Domain-specific languages
Systems architecture	Infrastructure definition Automated deployments
Operations	Service level agreement Monitoring

models becomes apparent. Furthermore, the degree to which goals are met needs to be measurable. This calls for a business model to be formulated at the very beginning of a model-supported software creation project.

A project starts with the identification of a problem to be solved as a contribution to a business goal. In many cases, the problem does not lie within the computing domain. Accordingly, the desired solution is typically formulated by means of (project) goals that shall be reached.

Goals have to be measurable in order to judge the success of a project. *Key Performance Indicators* (*KPIs*) or *Objectives and Key Results* (*OKRs*) are used to define target values. The values that are measured often lie in the business domain and have to be determined by controlling means on the business level. The success of a software solution that helps reaching the goal is then proven implicitly, assuming that it substantially contributes to reaching the business goals.

Since formal goals are set up as a first abstraction of the business goals to be reached, they are subjective and depend on a stakeholder who defines them. Approaches like i* [6] aim to model this subjectivity.

B. Requirements

Requirements characterize the properties of a software solution. This means that this stage only is entered if it is decided that software helps reaching the defined goals. It also means that a first software solution hypothesis has been recognized and is being detailed through requirements.

There is a wide range of requirements: functional requirements and the diverse kinds of non-functional requirements. Together, they form a first model of a software solution that, however, is typically informal.

Additionally, (project) constraints that limit the solution space belong to this stage.

There are various tools to help managing functional requirements. Deductive databases can help validating and completing requirements [7].

C. Subject Domain Model

The later stages of software design require a certain understanding of the problem domain, for example, typical concepts of the area the software is to be applied in. The requirements relate to the domain concepts.

Modeling means abstracting from the domain that is represented. Therefore, domain concepts cover a section of the subject domain that is relevant for the solution.

In the MDA approach, the CIM may include the stage of domain modeling.

D. Creative Tasks in Software Development

Requirements can be defined in different ways. In requirements engineering, one aims at specifying properties of the software to be built with adequate precision. In agile approaches, requirements are formulated from a business perspective.

The formal models underlying the model-driven approaches as discussed in Section II require abstraction capabilities for viewers to imagine the software to be built and how it will meet the requirements. The ability to work with such abstractions cannot be taken for granted for domain experts and various stakeholders in a software project.

Modern development approaches, in particular agile ones, are based on the engagement of stakeholders, though, requiring them to understand the outcome of each development step. To this end, often visual communication is used for participants who are not comfortable with working with abstract representations of software as used by software developers in an MDSE process. This visual communication is based on the creative input contributed by a cross-functional team. Such creative input is usually found in software development processes with a high degree of creativity involved, like, for example, solutions with a focus on human-machine interaction or content-centric applications.

Creatively working team members and certain other domain experts prefer using specific notations and tools. Creative tasks are typically carried out on the basis of visual presentations that lack formal completeness and soundness. Still, they help develop a common understanding between the various participants in a project and are, therefore, central to the communication with stakeholders and domain experts to allow them to judge the ongoing development on a subjective basis.

Formal models are well-suited to reason about the emerging software solution, but not about the inputs that led to its design.

For all these reasons, informal means of communication between creative workers, analysts, etc. and stakeholders in a *user-centered development process* typically cannot be substituted by abstract models of software alone.

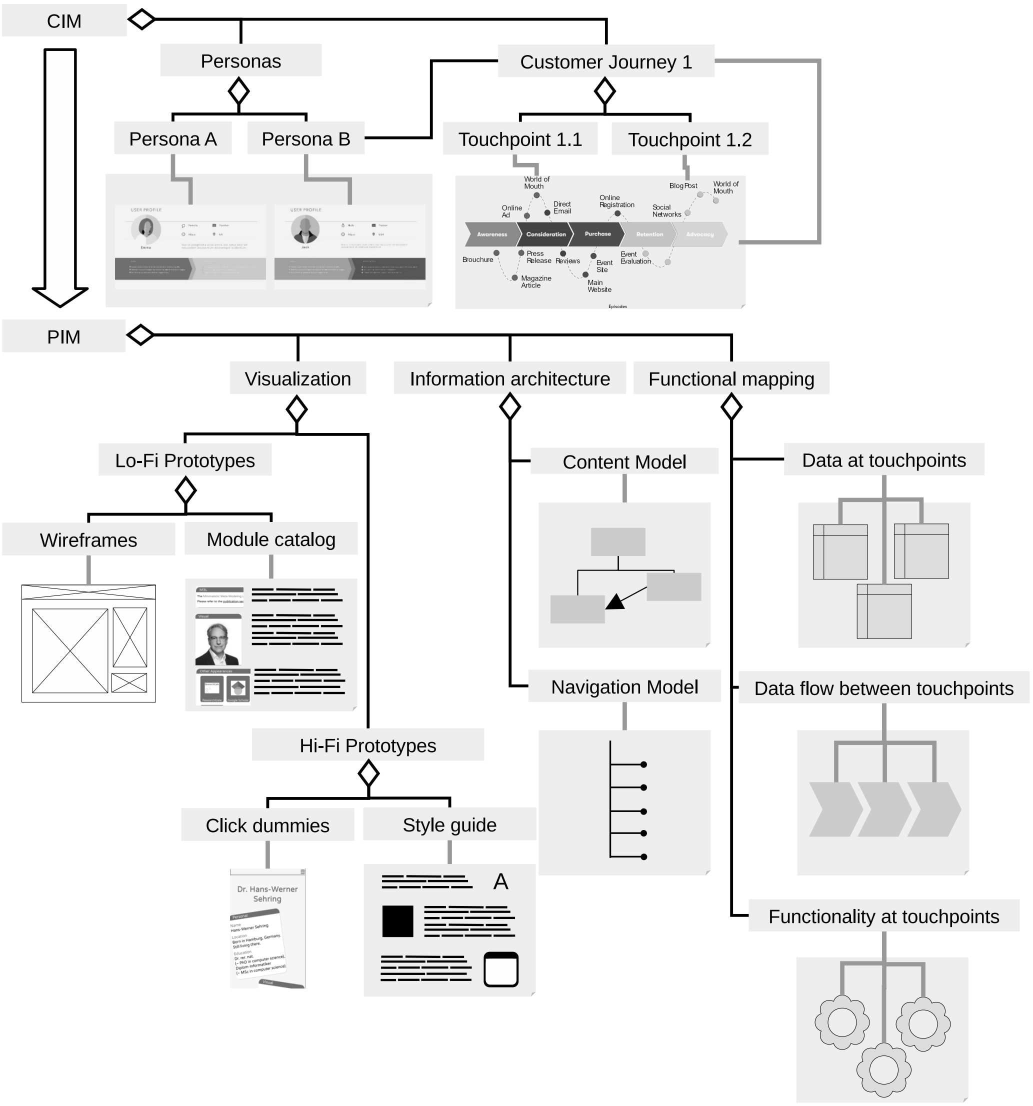


Figure 1. Creative tasks in software development processes.

In MDA terms, a CIM is elaborated to a "visual PIM" that describes the software from the perspective of users (in contrast to the developers' perspective). In this PIM, there is an emphasis on the user experience and the visual appearance of software. This model defines how users of a service interact with the provider of that service, for example, how customers interact with an online shop.

Figure 1 sketches a typical user-centered development process, with the sequence of models shown at the left of the figure.

Conceptions of interactive applications for digital communication typically begin by identifying personas as role models of target groups, determine the customer journeys as the sequence of interactions users have at different touch points, before finally deriving artifacts like the information architecture.

To design user interfaces, artifacts like wireframes and sketches provide first impressions. Later, graphical details are added in documents like style guides. Click dummies allow stakeholders to get an impression on how the software is intended to be operated.

All these conceptual and graphical descriptions are, on the one hand, defining user experience aspects like navigation and visual layout. On the other hand, they contribute to the definition of subject domain concepts and requirements.

This phase of creating a visual PIM is not concerned about software implementation. Feasibility studies for the realization of the design in software should guide the creative activities, though.

Software design is added at a subsequent stage by creating a PIM in the sense of the MDA. Starting from this point, the steps of typical MDSE processes follow.

The transition from a visual, user-centric PIM to a software-centric PIM depends on the descriptions used. If the aforementioned typical results like personas and customer journeys are created, one step is to identify the information need of users at each touch point, and the resulting data flows that are required to fulfill the information needs. Likewise, the overall functionality is broken down to functionality of each touch point that results from the interface conception.

E. Solution Architecture

Solution architecture is the set of high-level definitions that relate subject domain concepts to technical solutions.

As a high-level architecture, it does not prescribe an actual implementation in full detail. It may contain the choice for certain implementation technologies and products, though, in particular if they are crucial to meeting some requirements or to conform to the constraints.

Based on the chosen components, a solution architecture defines the interfaces required to implement the processes and data flows identified as requirements. For example, in a digital communication like an e-commerce website, the information demand at every touch point is derived from the customer journeys, and data flows are designed accordingly.

F. Software Architecture

The detailed design of the software to be developed is part of the software architecture. It details definitions from the solution architecture up to the point where they are concrete enough to guide the coding stage.

Shaw and Garlan [8] point out that there are different approaches to the different perspectives on software. In a structural approach, the software architecture is composed of components, communication between the components, product configurations, references to the requirements and constraints from the requirements stage, boundaries within which the software is designed to work as specified, the rationale of design decisions, and design alternatives that were considered.

Many other architecture definitions contain similar modeling entities. *Architectural Description Languages* (*ADLs*) allow capturing these aspects.

Shaw and Garlan point out that besides structural models, there are also framework models, dynamic models, and process models. The latter, for example, focus on the dynamic aspects of the software.

G. Code

When architecture models are precise enough, code can be generated out of them using one of the approaches from Section II-B.

In practice, coding is a manual task in most cases. The architecture definition serves as a guideline to programming, documentation, and quality assurance. Detailed design decisions are added in the coding stage.

There may be another modeling step included, though. Software may be defined in an abstract way, close to programming but abstracting from concrete programming languages and other base technologies. Software is derived from such abstract code bases by means of code generation. This helps building multi-platform software and avoiding repetitive coding tasks. This topic is revisited in Section VI.

H. Systems Architecture

The systems architecture describes how software is deployed and set up. It defines computing and communication infrastructure.

Deployment diagrams describe how software is packaged and distributed on the infrastructure. Infrastructure and network diagrams illustrate the technical setup.

Typically, infrastructure is virtualized and created automatically from scripts in the *Infrastructure as Code* approaches. This allows continuous deployments of many software components, for example, in contemporary composable architectures.

I. Roll-out and Operations

MDSE processes are primarily concerned about the creation of a software solution. The overall software lifecycle requires the consideration of further phases. Modern development approaches take these into account, for example, by providing product increments in agile approaches and through DevOps methods.

Consequently, model-supported processes should consider project stages after software generation in due time, namely roll-out, operations, maintenance, and support. Optimally, there are explicit descriptions of the activities in those phases and for the precautions to be taken by software.

For software roll-out, for example, deployment scripts can be generated for software installation. On a higher level, and International roll-out mean orchestrating a global team of people in different roles. The orchestration needed is partly dependent on the solution design.

Other activities of a software roll-out are concerned about establishing user-acceptance, for example, by providing documentation and training. Documentation and training are, of course, dependent on the software and may partially be generated from the software models.

When it comes to operations, in particular in the presence of virtualized, eventually cloud applications that are common today, software is built in a way that allows utilizing the advantages of the operations infrastructure. For example, scalability and elasticity have to be considered in software architecture. This way, software design decisions made early in the process have have an impact on the operation of the software. For this, the models of early software design contribute to operations models.

Also, software needs to meet non-functional requirements like maintainability for the operations phase. To this end, for example a logging concept needs to be considered in the MDSE models. Remote logging is particularly important in distributed systems, for example, incorporating mobile apps.

Part of the requirements are typically formulated towards operations. *Service-Level Agreements* (*SLAs*) define measurable goals to systems operation. Fulfillment of these goals is controlled by means of monitoring and timely maintenance in the case of incidents. To this end, monitoring and logging concepts connect development and operations.

	Data model	**Progr. language**	**OO model**	**Software model**
Metameta Layer	Data modeling style	Syn.+sem. definitions	Class *MetaClass*	M³L
Meta Layer	Data definition lang.	Programming lang.	MetaClasses	Metamodel
Abstract Layer	Schema	Programm	Classes	Software model
Concrete Layer	Data	Execution	Instances	Software solution

Figure 2. Modeling layers.

IV. MSSC INCLUDING NON-FORMAL ARTIFACTS

The integration of different kinds of descriptions into the models on each modeling stage is a second respect in which we feel that existing MDSE approaches need extension.

Descriptions range from software models with defined semantics and other formalisms to unstructured media like written text and images.

A. Formal Software Models

Existing MDSE approaches usually use formalisms that allow assigning formal semantics to models, and to apply model transformations. The formal models that are used depend on the utilization of the models, for example, UML diagrams if there is an emphasis on software architecture, as in the MDA, or Petri Nets to model behavior [9].

While these models are well-suited for MDSE approaches as they allow model transformations, they are primarily appealing to those with a formal background, for example, computer scientists. Therefore, formalisms provide a sound basis for the construction of, for example, scientific and engineering applications. But experts in other domains may be less comfortable with providing abstract models of their domain and the desired software solution.

B. Visual Descriptions of Software

MDSE typically is based on a modeling framework that supports all stages of a software development process. This requires that model artifacts on every stage can be expressed in a language that is supported by that framework. In many cases, it is even required that all models involved are formulated within the same metamodel.

Some application domains call for specific kinds of artifacts that rely on certain established notations and that cannot be expressed in the form of a given central model. For such application domains, the properties of software are designed by experts who use specific notations and tools. Digital communication like marketing and sales communication over a website is an example of such an application domain.

In the retail sector, for example, we note that customers interact with retail companies at different touch points, interact on changing communication channels, use different payment methods, are subject to different legal and tax systems, etc. In such scenarios, a series of experts needs to gather (a part of) the domain knowledge on one modeling stage in order to communicate it to experts of the next stage (domain expert to requirements engineers, these in turn to architects as well as test engineers, architects to developers, and so on).

User experience designers and user interfaces designers, for example, work with artifacts like personas, customer journeys, wireframes, style guides, click dummies, prototypes, etc. Such artifacts support creative processes. They are adequate means to communicate with business experts, and they are used by programmers to build usable software.

A pure MDSE approach of generating such artifacts from models is not adequate for the work of experts and their clients. It might be hindering the creative process.

C. Metamodeling

The heterogeneity of models that together build the basis for MSSC raises the question of how to relate different models to each other. Our answer to that question is a common modeling base that provides a framework in which different modeling approaches can be applied.

Modeling layers that build upon each other are found in various places in computer science. A stack of four modeling layers, where the topmost layer is recursively defining itself, is found in various places. The layers range from a layer of concrete entities to a meta meta layer in these cases [10]. Figure 2 illustrates this.

The four levels of modeling are found in database models, for example, where data are on the concrete layer. Data is described by a data model or schema, that in turn is given in some data definition language. A data definition language is formulated with respect to a data modeling style, for example, the relational data model.

Also, we see for layers in programming language, where a program is an abstract concept that is instantiated in a program run on the concrete layer. The program is written in a programming language (meta layer) that builds upon general notions of syntax and semantics definitions.

In object-oriented modeling, these layers may all be expressed within one programming language. Objects are defined by classes, classes are in turn defined by metaclasses that are instances of one common metaclass.

In software engineering, there are different perspectives on a software solution and the project in which it is created, and models from different perspectives are formulated in different notations. This means that there are different languages (aka metamodels) for the co-existing perspectives that need to be integrated in one holistic modeling process. This calls for a meta-meta-model which establishes a common ground on which metamodels are defined.

The rightmost column indicates that the modeling language used for experiments in this article, the M³L, also fits into the pattern. Being a metamodeling language, it is itself to be located at the metameta level. We chose the M³L because it allows to be applied on all levels and it does not differentiate between relationships on one level (for example, subtyping or aggregation) and relationships between level (for example, instantiation). More on this in the subsequent section.

D. *Model Refinement and Transformations*

An MDSE process relies on a series of models where models are created from existing models by means of *model transformation*. A model on one stage is created based on the input of models of earlier stages or by refining models from the same stage. There are three typical kinds of model transformations.

Figure 3a shows the basic structure of model transformations on one stage and between stages. Figures 3b to 3g show examples of typical model transformations between different stages.

a) Model Combination: Domains often rely on base domains. For example, business tasks rely on mathematics. Accordingly, models are defined by integrating (existing) models of the base domains. This way, models are reused.

b) Model Refinement: Within one stage, models are refined to more concrete models of the same stage. This way, the work in each stage starts with first, coarse-grained models, that are then transformed into more concrete models. Different refinements of one model may cover different perspectives on the (software) solution. The process of refining involves decision making. Decisions can be documented by explicitly stating delta models that explicitly represent the refinements.

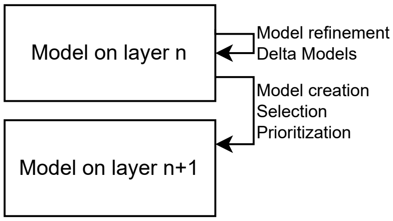
c) Model Creation from Existing Models: When processing from one stage to another, initial models are required for the subsequent stage that is entered. These models shall be related to the most concrete models of the preceding stage. In some cases, models can be transformed when proceeding to a subsequent stage. In this case, the transformation establishes the relationship. If new models have to be created, the model elements should be explicitly linked to the elements from models on which they are based. For example, Shaw and Garlan [8] demand that a software architecture description refers to requirements.

V. A BRIEF INTRODUCTION TO THE M³L

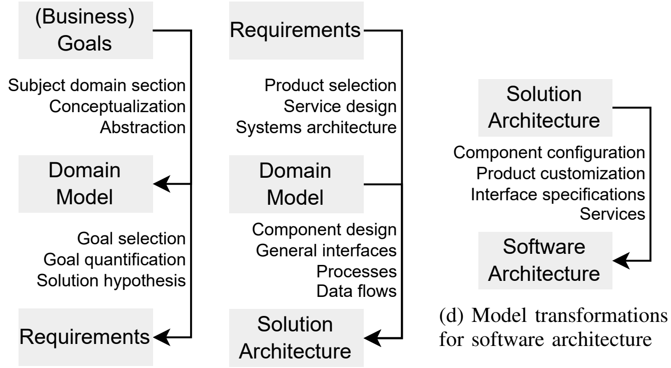
In this paper, we propose using the *Minimalistic Modeling Language*, *M³L* (pronounced "mel") [11], as the modeling framework required for MSSC.

The M³L is a meta modeling language. As such, it can be employed for models for different kinds of applications.

In this section, we give a brief overview over the syntax of the language. Sample applications in the subsequent sections demonstrate its use.



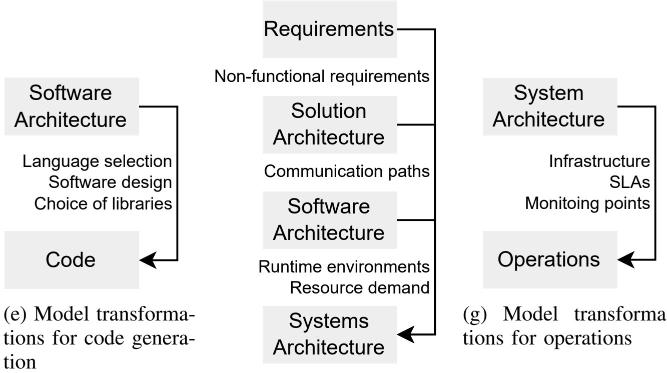
(a) General model transformations



(b) Model transformations for subject domain model

(c) Model transformations for solution architecture

(d) Model transformations for software architecture



(e) Model transformations for code generation

(f) Model transformations for system architecture

(g) Model transformations for operations

Figure 3. Different kinds of model transformations.

A. *Basic Definitions*

A M³L statement

```
A
```

defines or references a *concept* named *A*. The M³L does not distinguish definitions from references. If **A** does not exist, it is defined.

Concepts can be *refined* with "is a":

```
A is a C
```

Using the clause "is the" defines a concept to be the only specialization of its base concept.

Concepts can be put in *context*. A statement

```
A { B }
```

defines *B* in the context of *A*. *B* is said to be the *content* of *A*. References are valid in the context they are defined in and in all subcontexts. This means, that statements

```
A { B }
C
```

make *B* and *C* visible in the context of *A*, but *B* is not part of the content of *C* or of the topmost context.

Concepts can be defined differently in different contexts. For example, the statements

```
A { B is a C }
B
```

define *B* as a specialization of *C* in the context of *A*, and without base concept in the topmost context.

A concept in a nested context is referenced as

```
B from A
```

B. Concept Evaluation

Semantic rules can be defined on concepts, denoted by “|=”. A semantic rule references another concept that is delivered when a concept with a semantic rule is referenced. Like for any other reference, a non-existing concept is created on demand.

Context, specializations, and semantic rules are employed for *concept evaluation*. A concept evaluates to the result of its syntactic rule, if defined, or to its *narrowing*. A concept *B* is a narrowing of a concept *A* if

- *A* evaluates to *B* through specializations or semantic rules, and
- the whole content of *A* narrows down to content of *B*.

To evaluate a concept, syntactic rules and narrowing are applied repeatedly.

With this evaluation, for example, a conditional statement as found in imperative programming languages can be defined as (given *Statement*, *Boolean*, *True*, and *False*):

```
IfThenElseStatement is a Statement {
  Condition     is a Boolean
  ThenStatement is a Statement
  ElseStatement is a Statement }
IfTrueStmt is an IfThenElseStatement {
  True is the Condition
} |= ThenStatement
IfFalseStmt is an IfThenElseStatement {
  False is the Condition
} |= ElseStatement
```

A concrete program derives a conditional statement from *IfThenElseStatement*:

```
Conditional17 is a IfThenElseStatement {
 SomeBoolExpression is the Condition
 SomeStatement      is the ThenStatement
 SomeOtherStatement is the ElseStatement}
```

When evaluated, such a conditional statement will match (become a derived subconcept) of either *IfTrueStmt* or *IfFalseStmt*, depending on the concept that *SomeBoolExpression* evaluates to. from the derived base concept, the corresponding semantic rule will be inherited, making the statement evaluate to either then “then branch” or the “else branch”.

Concepts are evaluated with respect to an evaluation context. Concept definitions that contribute to the evaluation of a concept are taken from that context.

C. External Concept Representations

Concepts can be marshaled/unmarshaled as text by *syntactic* rules, denoted by “|-”. A syntactic rule names a sequence of concepts whose representations are concatenated. A concept without a syntactic rule is represented by its name. Syntactic rules are used to represent a concept as a string as well as to create a concept from a string.

For example, rules for language-dependent code generation can be given as:

```
Java {
  IfThenElseStatement
  |- if ( Condition )
     ThenStmt
     ElseStmt . }
```

In this example, an *IfThenElseStatement* will be used to generate Java code when it is marshaled in the context of the concept *Java*.

Not that the concepts *if*, *(*, and *)* are created in this syntactic rule. Since every concept, by default, represented by its name, these concepts can be used like string literals.

VI. AN MSSC APPROACH WITH THE M³L

An MSSC approach includes the creation and utilization of diverse artifacts. Each of them serves a specific purpose, and each is maintained by experts using established tools. Though the artifacts from different stages of a software creation process are related, they typically cannot be expressed using the same language. They differ, for example, in the level of detail, the degree to which they follow a formalism, and the syntactic representation targeted at different audiences.

When, in contrast to MDSE, no single modeling language can be used for a universal model, an overarching modeling framework is required for model coherence [12]. Such a framework cannot host the artifacts themselves. It shall, however, put the artifacts in context and relate them to each other.

Relationships between artifacts clarify their contribution to the software creation process. They explicate the provenance of models, they put models in context, and they are the basis for traceability and, therefore, the ability to cope with change.

The three model relationships named in Section IV-D can be expressed with the M³L. This way, models are put in context.

Also, code can be generated from M³L models.

A. Combining Models

Let *BaseModel1* and *BaseModel2* be some models of some domains whose concepts can be reused for the domain at hand. Then, for example, concepts *A* and *B* can be integrated into a new model *SomeModel* by definitions like

```
SomeModel {
  A from BaseModel1
  B from BaseModel2 }
```

For example, on the layer of domain models, a model

```
ProductDescriptions is a DomainModel {
  ProductData
  PaymentMethods        from Commerce
  PackagingInformation  from Logistics }
```

combines parts of product details that come from different specialized models (assuming that concepts for models *Commerce* and *Logistics* are given).

Likewise, on the layer of solution architecture, a model

```
OurInfoSys is a PlatformIndependentModel {
  AppServer   from SWComponents
  DBMS        from SWComponents
  DataSchema  from DBModeling
  WebServer   from SWComponents
  WebPage     from WebDesign }
```

combines technical components from different technical descriptions.

B. Refining Models

One model can be created as a refinement of another. Concepts in the content of the refined model are inherited and can be refined further.

```
SomeModel { A { C } }
```

can be refined to

```
RefinedModel is the SomeModel {
  A is a D {
    C is an E }
  B }
```

Making the *RefinedModel* the only specialization of *SomeModel*, all references to *SomeModel* are then narrowed to *RefinedModel*.

An example from the solution architecture layer is:

```
OurInfoSysConcept is an OurInfoSys {
  RDBMS from SWComponents is the DBMS
  ProductDataSchema
    is an RDBSchema from DBModeling,
       the DataSchema
  WebServer from SWComponents
    is a ServletEngine from Java }
```

In this example, two aspects of the conceptual model are refined: From a technical perspective, the *DBMS* is more concretely specified to be a relational DBMS (*RDBMS*), and the *WebServer* to be implemented as a Java Servlet engine (*ServletEngine*). Regarding the domain model, it is defined that the data schema is defined to store products (*ProductDataSchema*).

C. Creating Models of a Subsequent Stage

A model can be explicitly created as a transformation of another model using a semantic rule. For example, a model *RefinedModel* on a modeling stage *Stage1*,

```
Stage1 {
  RefinedModel { A } }
```

can be amended with a semantic rule to produce a model in a subsequent stage *Stage2*:

```
Stage2 {
  RefinedModel |= SomeDerivedModel {
    F is an A {
      G is the C }
    H {
      I } } }
```

This way, the model *SomeDerivedModel* is connected to *SomeModel* by the semantic rule, as it is its whole content. The concept *B* is not considered in the derived model.

In the example of the information system:

```
OurInfoSysConcept |= OurInfoSysDataLayer {
 RDBMS
 ProductDataSchema {
  ProductsTable is a Table from DBModeling
 } }
```

RDMBS from the source model *OurInfoSysConcept* is reintroduced in the transformed model. The database schema *ProductDataSchema* is additionally redefined by naming one table. *WebServer* from *OurInfoSysConcept* is not considered in the transformed model, since it only models the data layer of the information system.

D. Software Creation with the M³L

The models in MDSE ultimately reach the stage of generating code. The M³L allows creating code using syntactical rules that can be added to models with sufficient concreteness.

A simple example of Java code generation is shown in Section V-C.

Using the example from above, part of the information system based on a relational database can be defined to create a relational schema by SQL statements as follows:

```
OurInfoSysDBIm is an OurInfoSysDataLayer {
  ProductDataSchema {
    ProductsTable
    |- PRODUCTS( Columns ) .
    ProductSKUColumn is a Columns
    |- STOCK_KEEPING_UNIT VARCHAR(50), .
    ProductNameColumn is a Columns
    |- NAME VARCHAR(100), . }
 |- "CREATE TABLE " ProductsTable . }
```

By defining the syntactical rules in the context of an implementation model, different code generation schemes can be defined for one software model.

In the example, in the context of a different implementation model, syntactic rules for the generation of database access code may be defined on the concepts like *ProductDataSchema* and *ProductData*.

E. Metaprogramming with the M³L

Instead of generating code directly by syntactic rules, code can first be modeled in an abstract way as indicated in Section V-B by the example of the if…then…else statement. This allows to consistently generate code in multiple languages.

For example, consistency of table and column names in data definition code and data access code is achieved by using the same concepts during code generation:

```
OurInfoSysDBIm
  is an OurInfoSysDataLayer, an SQL {
  ProductDataSchema is a Schema {
    PRODUCTS is a Table {
      STOCK_KEEPING_UNIT is a Colum
      NAME               is a Colum } } }
OurInfoSysDBAccess
  is an OurInfoSysDataLayer, a Java {
  Product is an Interface { ... }
  ProductImpl is a Class {
    Product is an Interface
    retrieve is a Method {
      ... SELECT * FROM
      PRODUCTS from OurInfoSysDBIm
      WHERE ... } } }
```

In this example, the table *PRODUCTS* is referenced as a concept, guaranteeing that the table and column names included in the Java code are identical to the ones used in the SQL data definition statements.

The syntactic code generation rules are inherited from the concepts defined in the code models, *SQL* and *Java* in this example.

This way, we derive a PSM and finally code by means of model transformation, where the syntactic rules are coming from language models / metamodels. This meta level aspect can be found even more explicitly in the GraSyLa [13], for example.

VII. CONCLUSION

This section sums up this paper and outlines future work.

A. Summary

In this paper, we revisit MDSE approaches and conclude that they are successful in certain application areas, while they are not established in many other areas. In particular, in digital communication, for example, in the construction of commerce or marketing websites or mobiles apps, they are not used in practice. One reason for this is a mismatch between established means of conceptual work and formal models.

Under the name of Model-Supported Software Creation (MSSC) we study requirements to models for such kind of applications. As early results, MDSE approaches cover the stages of software creation well, but they do not cover early inception phases. We claim that models used in MSSC need to be able to cope with less formalism and preciseness as required by typical MDSE approaches. Instead, they must deal with heterogeneity and subjectivity.

We outline model creation with the M³L as a step towards MSSC. It allows providing descriptive models of the artifacts used in practical approaches and relating them as to drive holistic software creation processes.

B. Outlook

We are at the beginning of our investigations towards MSSC. Consequently, there are numerous questions to be answered in the future. We highlight two of them.

There are numerous approaches to generate code from models, and code written in a formal language can be managed in a structured way. The syntactic rules of the M³L, for example, allow this. To include artifacts from other stages into the modeling process (like requirements or design documents), abstractions are needed to reference, include, or generate parts of artifacts the same way it is possible for code.

Testing is typically not found in model-based processes. Though there may be no need to test generated software, a kind of testing is required, nevertheless. This may include model checking on each stage of the process and analysis of models that are the result of model transformations.

In MSSC processes, success should ultimately be judged based on the degree to which business goals have been reached. To this end, they must be formalized, and effects of the running software need to be measured.

ACKNOWLEDGMENT

Numerous discussions on topics of modeling and software engineering led with colleagues, partners, and clients are highly appreciated.

The author thanks the Nordakademie University of Applied Sciences for funding the publication of this work.

REFERENCES

[1] H.-W. Sehring, “Model-supported Software Creation: Towards Holistic Model-driven Software Engineering,” Proc. Eighteenth International Conference on Software Engineering Advances, 2023, pp. 113-118.

[2] J. Cabot, R. Clarisó, M. Brambilla, and S. Gérard, S., “Cognifying Model-Driven Software Engineering,” Proc. Software Technologies: Applications and Foundations (STAF 2017), Springer, 2018, pp. 154-160.

[3] Object Management Group. *Model Driven Architecture (MDA)*, MDA Guide rev. 2.0, OMG Document ormsc/2014-06-01, [Online] Available from: https://www.omg.org/cgi-bin/doc?ormsc/14-06-01. 2024.3.11.

[4] H.-W. Sehring, S. Bossung, and J. W. Schmidt, “Content is Capricious: A Case for Dynamic System Generation,” Proc. 10th East European Conference (ADBIS 2006), Springer, 2006, pp. 430-445.

[5] T. Kühne, “Matters of (Meta-) Modeling,” Software & Systems Modeling, vol. 5, pp. 369-385, Dec. 2006.

[6] E. S. K. Yu and J. Mylopoulos, “From E-R to “A-R” – Modelling strategic actor relationships for business process reengineering,” Proc. 13th Int. Conf. on the Entity-Relationship Approach (ER’94), Springer, 1994, pp. 548-565.

[7] H. W. Nissen, M. A. Jeusfeld, M. Jarke, G. V. Zemanek, and H. Huber, “Managing multiple requirements perspectives with metamodels,” in IEEE Software, vol. 13, no. 2, pp. 37-48, March 1996.

[8] M. Shaw and D. Garlan, “Formulations and Formalisms in Software Architecture,” Computer Science Today: Recent Trends and Developments, Lecture Notes in Computer Science, vol. 1000, pp. 307-323, 1995.

[9] R. Koci and V. Janousek, “Prerequisites for Simulation-Based Software Design and Deployment,” Proc. Eighteenth International Conference on Software Engineering Advances, 2023, pp. 105-109.

[10] M. Jarke, Matthias, R. Klamma, K. Lyytinen, “Metamodeling,” in Metamodeling for method engineering, M. A. Jeusfeld, M. Jarke, and J. Mylopoulos, Eds. Cambridge, MA : The MIT Press, pp. 43-88, 2009.

[11] H.-W. Sehring, “On Integrated Models for Coherent Content Management and Document Dissemination,” Proc. 13th International Conference on Creative Content Technologies (CONTENT 2021), 2021, pp. 6-11.

[12] S. Bossung, H.-W. Sehring, M. Skusa, and J. W. Schmidt, “Conceptual Content Management for Software Engineering Processes,” Proc. Advances in Databases and Information Systems, 9th East European Conference (ADBIS 2005), Springer, 2005, pp. 309-323.

[13] V. Englebert and K. Magusiak, “The Grasyla 2.0 Language,” Edition 1.2 (Draft), University of Namur, [Online] Available from: https://staff.info.unamur.be/ven/metadone.site/documents/techreport-grasyla-version-2.1.2.pdf. 2024.3.24.

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

An Integrated Software Manufacturing Control System for a Software Factory with Built-In Rejuvenation

Herwig Mannaert and Jan Verelst
Normalized Systems Institute
University of Antwerp, Belgium
Email: {herwig.mannaert, jan.verelst}@uantwerp.be

Koen De Cock and Jeroen Faes
Research and Development
NSX bv, Belgium
Email: {koen.de.cock, jeroen.faes}@nsx.normalizedsystems.org

***Abstract*—Software engineers have been attempting for many decades to produce or assemble software in a more industrial way. Such an approach is currently often associated with concepts like *Software Product Lines* and *Software Factories*. The monitoring, management, and control of such factories is mainly based on a methodology called *DevOps*. Though current DevOps environments are quite advanced and highly automated, they are based on many different technologies and tools. In this contribution, it is argued that more integrated software manufacturing control systems are needed, similar to control systems in traditional manufacturing. This paper presents a scope, overall architecture and prototype implementation of such an integrated software manufacturing control system. Several detailed scenarios are elaborated that can leverage such integrated control systems to optimize the operations, and improve both the quality and output of modern software factories. The preliminary findings and results of this control system are presented and discussed.**

Index Terms*—*Software Factories; Software Product Lines; DevOps; Control Systems; Evolvability.

I. INTRODUCTION

This article extends a previous contribution which was originally presented at the Eighteenth International Conference on Software Engineering Advances (ICSEA) 2023 [1].

The expression *"Software is eating the world"* was formulated in 2011 by Marc Andreessen [2] to convey the trend that many industries were being disrupted and transformed by software. And indeed, more and more major businesses and industries are being run on software systems and delivered as online services. These software systems include *Enterprise Resource Planning (ERP)* systems to design and manage the business processes, *Supervisory Control and Data Acquisition (SCADA)* systems to manage and control production processes in real-time, and *Manufacturing Execution Systems (MES)* to track and document the transformation of raw materials to finished goods, enabling decision-makers to optimize conditions and improve production output. As software systems become more pervasive to manage and control the end-to-end production processes in factories, it seems logical to have or create such control systems for the software systems themselves, i.e., systems to manage and control the building and assembly of software systems in the software factories. In this contribution, we explore the creation of such systems to manage and control software manufacturing and assembly.

The remainder of this contribution is structured as follows. In Section II, we briefly discuss software factories, the *DevOps* methodology, and situate our approach. In Section III, we describe the software factory used in this case study, and present a model for the assembly lines or units of such a factory. In Section IV, we discuss the scope, software architecture, and the implementation characteristics of the proposed manufacturing control system for software factories. We present various use cases and types of added value for such an integrated control system in Section V. Section VI discusses some preliminary results and findings of the continuous development of the system in the controlled environment. Finally, we present some conclusions in Section VII.

II. SOFTWARE FACTORIES AND DEVOPS

A. *On Software Factories and Reusability*

The idea to produce and/or assemble software in a more industrial way, similar to automated assembly lines in manufacturing, has been pursued for many decades. Such an approach is currently often associated with concepts like *Software Product Lines (SPLs)* and *Software Factories*, but can easily be traced back as far as 1968 to the article on *mass produced software components* from Doug McIlroy [3]. The concept of *Software Product Lines* has been extensively described by the Carnegie Mellon *Software Engineering Institute (SEI)* [4], and refers in general to software engineering methods, tools and techniques for creating a collection of similar software systems from a shared set of software assets using a common means of production. The characteristic that distinguishes software product lines from previous efforts is predictive versus opportunistic software reuse, as it stresses that software artifacts should only be created when reuse is predicted in one or more products in a well-defined product line [5]. The term *Software Factory* emphasizes the techniques and benefits of traditional manufacturing, and is for instance defined by Greenfield et al. as a software product line that configures extensive tools, processes, and content using a template based on a schema to automate the development and maintenance of

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

variants of an archetypical product by adapting, assembling, and configuring framework-based components [6].

The reuse of software artifacts seems crucial in contemporary efforts to realize the benefits of traditional manufacturing through software factories. Nevertheless, the systematic reuse of software artifacts is not a trivial task. Saeed recently argued that software reusability is not just facing legal issues, but methodological issues as well. Even when only reusing software to save time, and leverage off the specialization of other authors, the end-user must also have the technical expertise to search, adapt and merge these reusable assets into the larger software infrastructure [7]. We have argued in our previous work that software reuse is even more challenging, and impeded by some fundamental issues related to software evolvability [8] [9]. The sustained technological evolution leads to a continuous sequence of new versions and variants of the software artifacts that need to be reused. These new artifact versions often require changes in their usage that ripple through the entire software structure, causing an impact that is dependent on the size of the system, and limiting the evolvability of software systems [8] [10].

B. From DevOps to Integrated Control Systems

The aim of this contribution is to explore the creation of systems to manage and control the building and assembly of software systems in software factories, similar to SCADA or MES systems in traditional manufacturing. The main approach today in the software development and IT industry to control the building and assembly of software is a methodology called *DevOps*. Used as a set of practices and tools, *DevOps* integrates and automates the work of software development *(Dev)* and IT operations *(Ops)* as a means for improving and shortening the systems development life cycle [11]. It also supports consistency, reliability, and efficiency within the organization, and is usually enabled by a shared code repository or version control. As DevOps researcher Ravi Teja Yarlagadda hypothesizes, *Through DevOps, there is an assumption that all functions can be carried out, controlled, and managed in a central place using a simple code* [12].

Figure 1 presents a traditional overview diagram of a typical DevOps infrastructure environment. While the continuous integration of the software development and IT operations is represented by the infinity symbol, the representation also contains a typical set of tools and technologies being used in such an infrastructure. We distinguish for example tools for tracking features and user stories (Jira), source control management (Git and Bitbucket), software quality control (SonarQube), automation of build pipelines (Jenkins), automated testing (Cucumber, JUnit), deployment infrastructure (Kubernetes), analytics visualization (Grafana), logging (Graylog), automated deployment (Docker, Ansible), and connecting cloud providers (AWS, Digital Ocean). While the tools in such a *DevOps* or *Continuous Integration Continuous Deployment (CICD)* infrastructure are in general numerous and versatile, there is a clear need for integrated control systems, similar to SCADA or MES systems, encompassing these processes and tools. However, software factories differ significantly from traditional industrial factories, as software is less tangible and the desired control systems need to interface with — often complex — software tools instead of physical equipment.

C. Related Work and Methodology

While academic research is available on various aspects of DevOps, like maturity assessment [13], and management challenges and practices [14], the development of integrated control systems does not seem to be one of them. DevOps platforms are considered to be based on a mix of open source and proprietary software, glued together and built into the platform by a platform team. At the same time, trade publications describe the necessity to breakdown the DevOps phases and tools to increase security and reduce technical debt [15], and acknowledge the need for solutions to scale up DevOps, as nearly a third of DevOps teams' time is spent on manual approaches that are not scalable [16].

The methodology of this contribution is based on *Design Science Research* [17], where we design the integrated control system for software factories as an artifact, and use a case study to evaluate it in depth in a business environment. Within the context of this case study, this contribution performs a controlled experiment, i.e., study the artifact in a controlled environment for qualities, in order to refine the artifact gradually as part of the design search process.

III. THE SOFTWARE FACTORY AND ASSEMBLY LINES

In this section, we describe the specific software factory that is used for the case-based design and evaluation of the software manufacturing control system. Within the context of this software factory, we present a model for an assembly line, a core concept in traditional manufacturing processes.

A. The NST Software Factory Case

To design and evaluate the integrated control system artifact, we use the case of *NSX*, the spin-off company that is developing and operating a software factory in accordance with *Normalized Systems Theory (NST)* [8] [9]. The software factory, described in detail in [18], encompasses both the metaprogramming environment, i.e., tools and code generators to *generate and rejuvenate* applications based on NST, and actual *Normalized Systems (NS)* applications, i.e., multi-tier web information systems generated in that environment. The various DevOps tools and technologies of the NSX factory correspond to a large extent to those in Figure 1. Though the company is limited in size, i.e., about 50 people, its DevOps environment supports the development and operations of a wide range of heterogeneous and interlinked software artifacts.

- *Run-time libraries* providing basic software utilities to various applications and tools.
- *Expansion resources* consisting of bundles of Normalized Systems code generation modules [9].

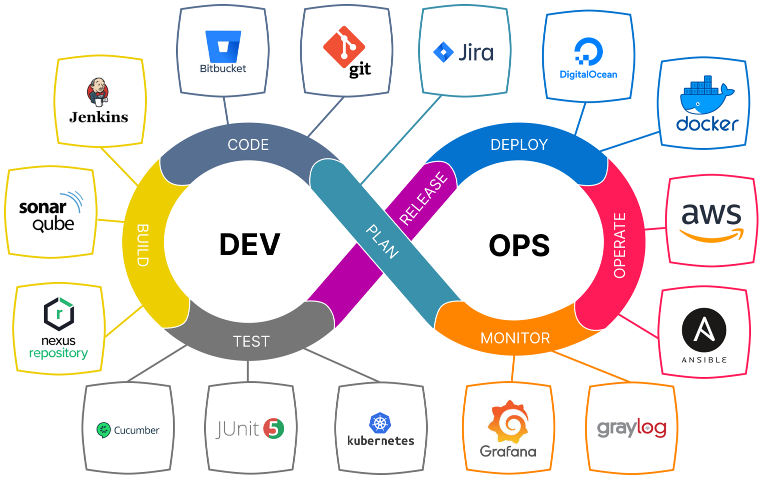


Figure 1. A traditional representation of a typical DevOps infrastructure.

- *Web Information Systems*, software applications based on the *Java Enterprise Edition (JEE)* standard.
- *Domain software components*, JEE components that are shared across multiple JEE applications.
- *Integrated Development Tool*, called *µRadiant*, to enable the model-driven development of NS applications.
- *Small tools and plugins* providing additional features in tools like *Maven* or *IntelliJ*, and the *µRadiant* itself.

The various build pipelines, defined in the corresponding software repositories, typically contain the following tasks.

- *Expanding* applications or components based on the NST metaprogramming environment.
- *Building* usable libraries, archives, or executables for components, applications, and tools.
- *Unit testing* of various software coding artifacts within the software repositories.
- *Reporting* on the repositories, such as test coverage or software quality metrics.
- *Deploying* live instances of applications or tools.
- *Integration testing* invoking live deployments.

B. A Model for Assembly Lines or Units

Assembly lines, as an implementation of the *Division of Labor* [19], are crucial to traditional manufacturing processes for mass production. To study and control a software factory, it is imperative to identify and model these basic building blocks that determine the sequential organization of adding and assembling parts. We propose to identify the fundamental assembly primitives of our software factory based on source code *repositories*. These repositories correspond to the units of work where programmers can contribute to the software, and are in general tightly connected to the automated DevOps build pipelines. We also propose to call them *assembly units*, as the number of steps or phases is limited, while the interactions with other units are diverse.

The architecture that we propose for such an assembly unit in the software factory is represented in Figure 2. The core of the unit is a repository, e.g., a *git* repository on *Bitbucket* or *Github*, with a corresponding configuration defining one or more build pipelines for an automation server such as *Jenkins*. Though not always present in every unit, we distinguish in general three activity phases in an assembly unit.

- *Expand phase*
  In the NST software factory, *expansion* or code generation of the skeletons precedes the compilation and building of the applications. Using modules or libraries of expanders, called *expansion resources*, this step is controlled by NS configuration settings that select (versions of) expansion resources. The automation server also performs a feedback cycle on the codebase, i.e., harvesting NS custom code and performing code analysis using tools like *SonarQube* and *Dependency-Track*.
- *Build phase*
  Compilation and building of the codebase is driven by a build automation tool like *Maven*, selecting through configuration the appropriate runtime libraries. The feedback cycle on the generated artifacts include an analysis of dependencies and vulnerabilities using tools like *Renovate* and *Sourcegraph*, and running (unit) tests.
- *Deploy phase*
  Generated artifacts that are executable can be deployed in containers using platform as a service tools and engines like *Docker*. This allows to perform live integration testing, possibly configured by tools like *Cucumber*.

The interaction or integration between the different assembly units is organized through a central repository of *executable artifacts* like expansion resources, runtime libraries, and application images. Assembly units can both retrieve these executable artifacts from, and submit them to, such an enterprise repository or *registry*. These registries are typically organized using tools as *Sonatype Nexus Repository* that offer additional functionality like access control. The reports generated by the various feedback cycles can also be stored and published on this central registry.

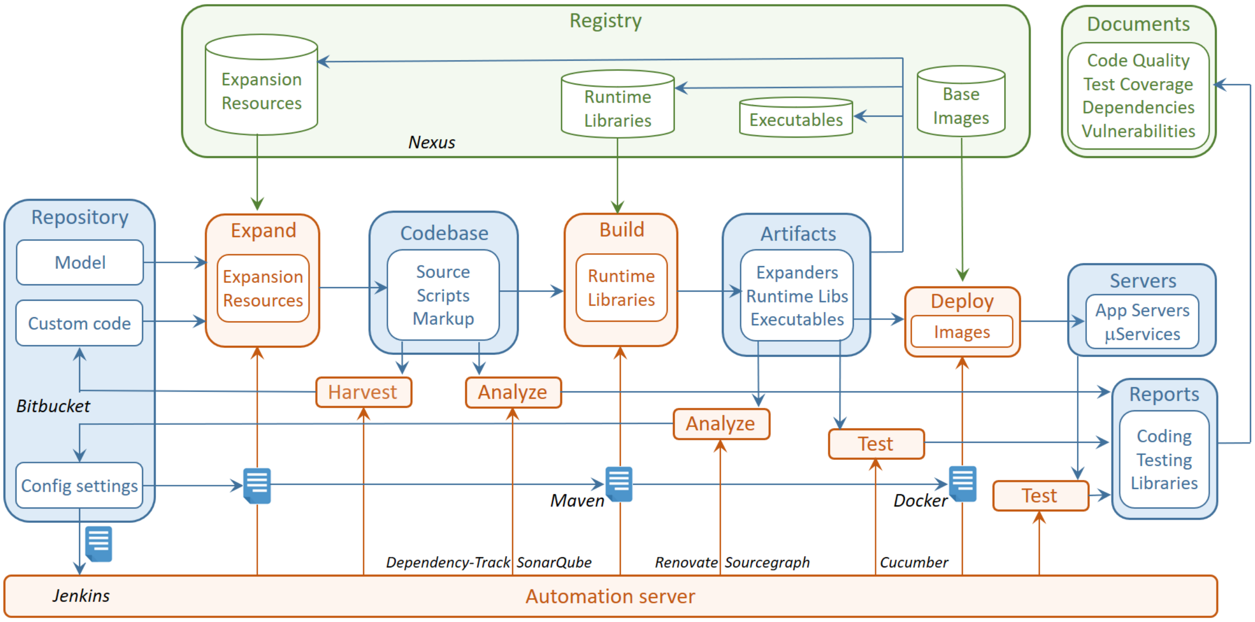


Figure 2. A representation of the architecture of a software factory assembly unit.

C. *Hierarchical Structure of Assembly Units*

The manufacturing of more complex products implies the existence of hierarchies of assembly lines. For instance, a car assembly line does not transform elementary screws and bolts into a car, but uses higher level modules like engines that are themselves produced in other assembly lines. In the same way, advanced digital platforms or applications cannot merely be built from simple source files, but use higher-level modules as well during their assembly. Explicitating these hierarchical assembly structures should be one of the features of the software manufacturing control system.

At the same time, not every phase or step is present in every assembly unit. For instance, assembly units for basic expansion resources or runtime libraries will neither have an expand phase, nor a deploy phase.

IV. A SOFTWARE MANUFACTURING CONTROL SYSTEM

In this section, we elaborate the purpose, scope, architecture, and implementation features of the software manufacturing control system, i.e., the artifact that is being designed and refined within the context of this case study.

A. *Purpose and Scope*

Consistent with the overall goal of software factories and product lines, many applications and tools in this DevOps environment are expanded and built by reusing and assembling various other software artifacts that are built in other repositories and pipelines. In order to have an idea of, for instance test coverage and code quality, in a certain version of a software application, we need an overview of these parameters across the versions of all the runtime libraries, expansion resources and components that are being used in that application. In other words, we need a *Software Bill of Materials (SBOM)*, and we want to be able to assess various parameters across this SBOM. Moreover, in the same way that manufacturers attempt to keep track of all the individual parts and ingredients that are part of delivered products, we want to track the various deployments, including the configurations of these deployment instances, for every version of every application.

This type of functionality, i.e., to manage and control end-to-end the building and assembly of software systems in software factories, is indeed similar to MES systems, i.e., to track and document the transformation of raw materials to finished goods, and SCADA systems, i.e., to manage and control real-time production processes, in manufacturing. And though almost all the required information is available somewhere in one of the various DevOps tools, the integrated overviews and aggregations are not easily accessible. This means that tracing specific parameters from various software parts to the deployed product instances requires in general manual effort in current state-of-the-art DevOps environments. Making this information instantly available is one of the goals of the proposed software manufacturing control system.

B. *Software Architecture*

The integrated software manufacturing control system artifact for the NSX software factory is implemented itself as a *Normalized Systems (NS)* application, allowing us to take advantage of the *NST* metaprogramming environment. Moreover, as NST was proposed to provide a theoretic foundation to build information systems that provide higher levels of evolvability [8] [10], NS applications are intrinsically suited for systems that need to integrate with various rapidly changing technologies and protocols. This should enable us to cope with the many, and rapidly changing, DevOps tools and technologies.

NS applications provide the main functionality of information systems through the instantiation of five detailed design patterns, termed *element structures* [8] [20]:

- Data elements to represent data or domain entities.
- Action elements to implement computing actions or tasks.
- Workflow elements to orchestrate flows or state machines.
- Connector elements to provide user or service interfaces.
- Trigger elements to trigger or activate tasks or flows.

At the core of every information system is its data model, consisting of the various domain entities and their relationships. A central part of the data model of our software manufacturing control system is represented in Figure 3. As every software artifact that is built resides in a *Repository*, we define an *AssemblyUnit* for every repository. For every assembly unit, automated *Pipelines* can be defined with different *PipelineVersions*. A first category of artifacts produced by assembly units are NS *Applications*, belonging to a *Domain*, and having different *ApplicationVersions* that can have various *ApplicationDeployments* themselves. The expansion of the NS applications is configured in *ExpansionResourceSettings*, and *ElementModelMeasures* and *CustomCodeMeasures* capture and track various measures of the application versions regarding both model and custom code. A second category of software artifacts are the expander bundles or *ExpansionResources* and their different versions. While the concept of an *ExpansionResourceDependency* is used to represent their mutual dependencies, various characteristics of these versions are stored in *ExpanderBundleMeasures*. A third category of software artifacts are (versions of) *DeveloperTools*. As these tools, including the *μRadiant* and various plugins, exhibit less predefined structure, we are limited here to capturing traditional software measures like technical debt.

The action or task elements serve to import, collect, and or compute various types of data for the software factory control system. Indeed, the manual entering of data in such a system would not only be extremely time consuming, it would also lead to consistency problems. More specifically, types of data that has to be collected, or computed, include:

- Versions of applications and developer tools with the corresponding versions of their dependencies.
- Aggregated information measures on NS applications, like the number of model entities, or the number and size of custom source code extensions and insertions.
- Aggregated information measures on NS expansion resources, like the number of individual expanders, or the number and size of expander templates.
- Overviews of automated tasks that have been performed in build pipelines with their result status.
- Various quality measures that have been computed for the various applications, expansion resources, and tools.
- Aggregated values for the use of various technologies, libraries and expansion resources in applications.

C. Implementation Features

A system or artifact for the monitoring and control of software manufacturing processes should be able to track the various parameters and data sets over time. This means that we need to track data over time for most data entities, like sizes of models and custom code, success rates of build processes, or software quality parameters. Therefore, the *history or log tables* are a crucial part of the data model. In the NS metaprogramming environment, *expanders* or code generators exist to automatically add —and even populate— an history element for every data element. These history tables can then be represented in graphs and analyzed over time, looking for possible improvements in productivity and/or output quality.

A large part of the relevant data for the software manufacturing control system is already present or computed in one of the many external tools or technologies represented in Figure 1. This implies that the automated collection and or computation of software factory data in automated tasks needs to integrate with these tools and technologies, such as *Bitbucket* repositories, *Maven* dependency declarations, *Jenkins* build engines, *SonarQube* code quality assessment, and *Dependency-Track* vulnerability analysis. In accordance with NST, there is a decoupling between the functionality of the data collection in the task element, e.g., build engine results or quality measurements, and the actual implementation (class) of the task element, e.g., getting data from *Jenkins* or *SonarQube*. In this way, the software manufacturing control system is able to support additional versions or variants of these tools and technologies with limited impact.

V. TOWARD A CONTROL LAYER FOR IMPROVEMENTS OF THE SOFTWARE FACTORY AND ITS OPERATIONS

As stated in Section I, by tracking and documenting the transformation of raw materials to finished goods, *MES* systems enable decision-makers to optimize conditions and improve production output. In the same way, a software manufacturing control system should provide an analysis platform and control layer to improve and optimize various aspects and characteristics of the software factory operations and output. In this section, we discuss some use cases and their added value, as they are being developed as part of the iterative case-based design process.

A. Monitoring Evolutions over Time

A first avenue to optimize and improve the output and quality of the software factory, is to monitor the evolution of certain parameters over time. As explained in [9], NS information systems distinguish between software skeletons, instantiations of element structures generated by modular code generators, termed expanders, and custom code being additional software artifacts or classes, i.e., *extensions*, or code snippets added to the generated artifacts or classes, i.e., *insertions*. From a quality and evolvability point of view, it is important to monitor the amount, size, and location of

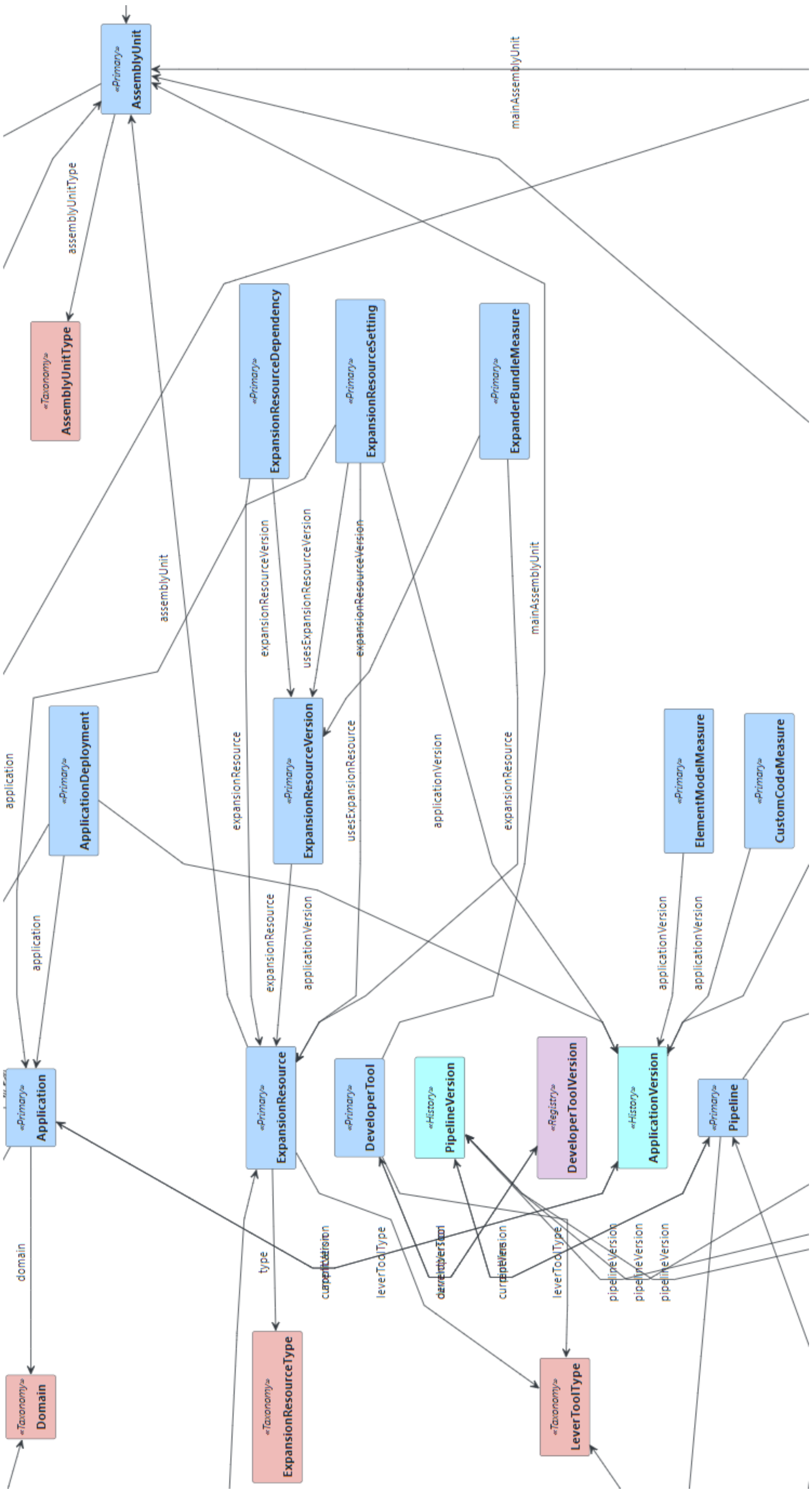


Figure 3. A representation of the data model of the integrated software manufacturing control system.

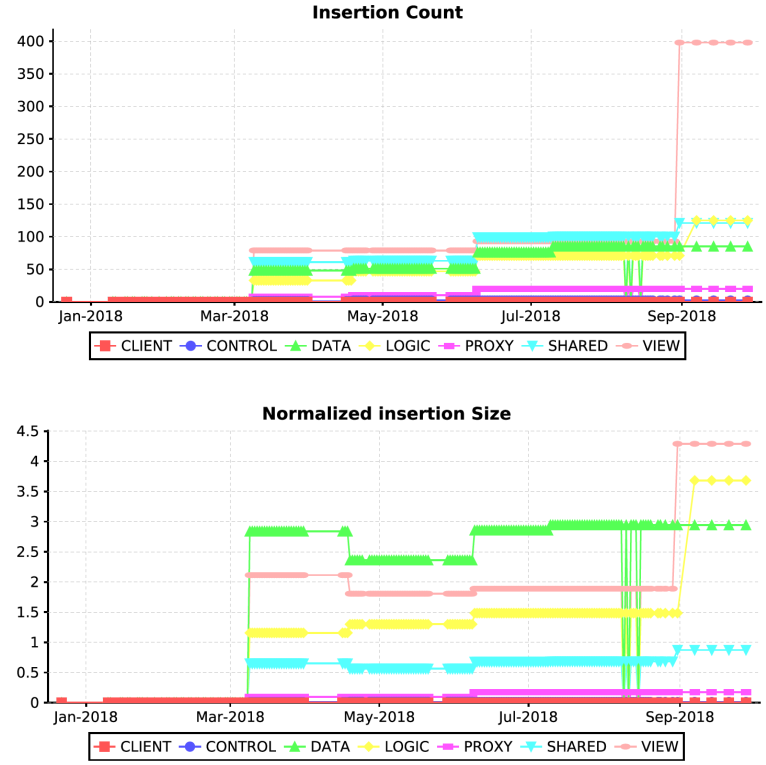


Figure 4. Sample graphs monitoring the amount of custom insertions and the normalized total size of insertions.

these extensions and insertions. As an example, some sample graphs are shown in Figure 4. They represent, for a specific information system, the evolution of the total amount of insertion snippets, and the total size of those insertion snippets. As shown in the second graph, these values can be made relative with respect to the evolving size of the model, i.e., the number of element structures. Different graph colors represent the various *layers* of the multi-tier web information systems. Higher levels of custom code can be specifically related to certain applications, but can also point to structural needs to provide additional expansion features in a specific layer.

This type of monitoring, based on automated data collection from the source code repositories, has been performed for quite some time in the NSX software factory. It has provided valuable insights into the actual project phases when such custom code typically grew fast, and the software layers where such custom snippets where needed the most. This last parameter provided an indication in which layer the development of the code generators should be prioritized to provide more out of the box functionality. Of course, other types of software systems and factories, without the typical NS distinction between element structures and custom code, should monitor other structural measures to improve software structure and productivity.

While this monitoring of custom code has been performed for some time, its integration in the overall software manufacturing control system provides additional added value. It will enable for instance a faster and broader analysis across multiple applications, comparisons between time periods and teams, and correlations with versions of expansion resources. Moreover, the monitoring over time is not limited to the source code repositories. History tables are also being created for success rates of build pipelines, quality measures of the custom source code, test overage percentages and numbers of failed tests, numbers of live system deployments, etc. Of course, this implies the integration with various DevOps tools and technologies, either through REST interfaces or reporting files.

B. Aggregating over Manufacturing Chains

It is often considered to be a crucial characteristic of software factories and software product lines that software artifacts should only be created when their reuse is predicted in one or more products [5]. And for instance in the NSX software factory, a typical JEE application uses various other software artifacts produced by the factory, such as:

- Several *runtime libraries* providing various utilities like file handling or protocol adapters.
- Several *reusable components* supporting more generic functionality like workflows or notifications, and/or providing more domain-specific building blocks, such as components for project planning or employee benefits.
- Several *expander bundles* that are used during the expansion of the application, such as the expanders to generate the instances of the NST element structures, or extensions such as the *Relational State Transfer (REST)* interfaces.

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

These artifacts are in general stored in other repositories and built in other CICD pipelines. And while the dependency on the code generation modules may be specific to NS applications, the dependencies on various runtime libraries and domain components are valid for nearly every software factory.

While parameters related to, for instance test coverage and code quality, can be monitored for every individual software artifact that is created in the factory, it seems quite relevant to offer instant overviews and aggregations of these parameters for all artifacts that are part of a specific aggregated artifact, such as a JEE application. While such integrated quality measures across a *Software Bill of Materials (SBOM)* are clearly relevant to the customers using or licensing such an application, this could also enable the optimization and improvement of the overall quality of the software factory itself [21]. Indeed, it would allow to identify the relatively weak or low quality parts in such aggregated artifacts, and to prioritize these software artifacts for improvements.

C. Tracking Technology Use Across Projects

Software applications are in general dependent on multiple *external* artifacts and technologies, e.g., libraries and plugins, that are build outside the software factory by commercial software vendors, or within open source projects. To address transparency around software applications, these components should also be part of a software inventory or SBOM. While these dependencies are available in configuration files, it is important to surface overviews and aggregations of these dependencies. Such overviews and their added value include:

- immediate overviews of the impacted applications when a vulnerability is detected in a library or technology.
- straightforward assessments of the impact when retiring a certain (version of a) technology.
- regular evaluations of the usage and adoption rate of libraries or expander bundles from the factory itself.

Obviously, such integrated information would also support decisions concerning internal resource allocation, both for supporting internal and external technologies.

VI. PRELIMINARY RESULTS OF THE CONTROL SYSTEM

The proposed software manufacturing control system aims to provide a transparent and traceable overview on the software factory, and to enable its steady improvement through a control layer. Though optimizations based on such a control layer takes time, we can already describe some basic results and findings from the initial implementation and data import.

A. On Software Applications

As mentioned in the previous section, the (versions of) NS software applications in this software factory have been analyzed on a continuous basis monitoring both the size of the models, i.e., data, task and flow elements, and the custom code, i.e., number and size of extensions and insertions. Therefore, it was pretty straightforward to import this data in the software manufacturing control system for the dozens of applications of the factory, corresponding to thousands of data elements.

Importing the expansion settings was relatively easy as well, and has already given insightful information, not just on the use of the various expansion resources, but also on the various versions that are being used at a specific point in time. This will clearly support the process of streamlining versions.

Constructing the SBOM or software inventory for external runtime libraries and technologies has proven to be less straightforward. As dependencies in *Maven POM (Project Object Model)* files are defined in a recursive way, we are currently investigating the integration of a suitable SBOM tool to flatten the resource and library dependencies.

B. On DevOps Infrastructure

Implementing the data import, we have observed a clear need for more structure and consistency across repositories and CI/CD pipelines. While some software applications were based on a single repository, others had one or multiple additional repositories, separating for instance authentication mechanisms, project-specific expanders, testing, master data, et cetera. This lack of consistency was also present in the definition of the pipelines in the various application repositories. Though other software factories may exhibit a superior consistency in the organization of the repositories and pipelines, consistent structure is something that typically arises during process automation. Therefore, it is quite possible that a structural improvement of this consistency could be a specific added value of a software manufacturing control system.

The integration of information from DevOps technologies like Jenkins and SonarQube needs to be performed carefully. As calling REST services to retrieve information from these tools could open up a path from the web-based manufacturing control system to those mission-critical servers, files are currently used to share information, and we are considering a callback architecture, where REST services on the control system are called from scripts in the external technologies.

C. On Expansion Resources

In the same way that model and custom code data have been imported for NS applications, retrieving data from (versions of) expander bundles or *expansion resources* has been integrated in the control system. Figure 5 presents for a number of expansion resources a schematic overview of the amount of expanders, i.e., the modular NS code generators, of expander features, i.e., feature modules within the expanders, and the total size of the templates (in bytes). Currently, the software manufacturing control system has imported (versions of) 66 different expansion resources. Besides showing that some expansion resources contain many heterogeneous expanders and need to be further modularized, this has also made clear that some additional streamlining and taxonomy is needed for the expansion resources.

Based on their configuration, the system also imports the dependencies between the (versions of) expansion resources.

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

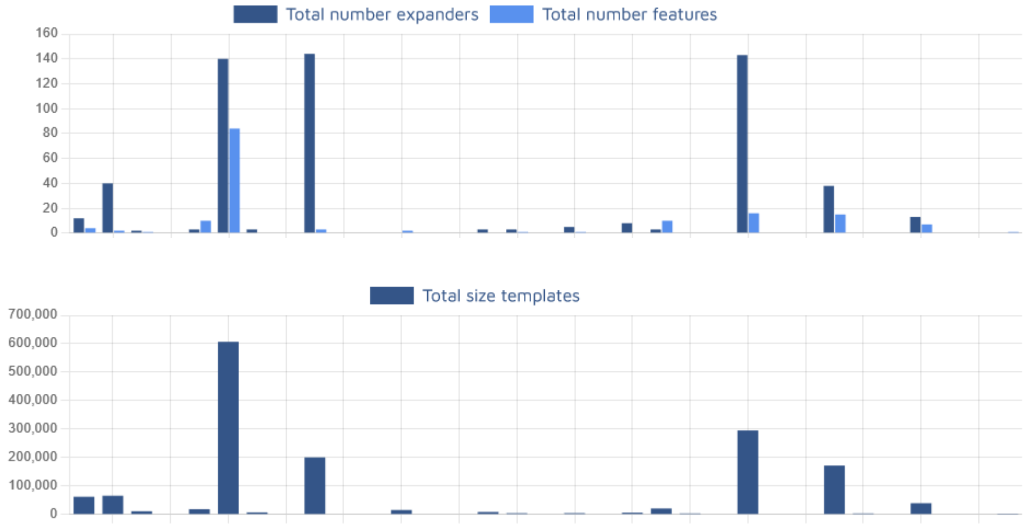


Figure 5. Sample graphs monitoring the amount of expanders and features and the total template size (bytes) of expansion resources.

As we represent and track these dependencies ourselves, the flattening of these dependencies to construct a SBOM has already been implemented.

VII. CONCLUSION

For many years, software engineers have strived to produce and/or assemble software in a more industrial way. In today's software factories, building and assembling software systems is mainly controlled using a methodology called *DevOps*. These *DevOps* environments are quite advanced and highly automated, but are in general based on many different technologies and tools. As previously experienced in the automation of business processes and traditional manufacturing, this often leads to a need for more integrated systems. In this contribution, we have investigated the creation of an integrated software manufacturing control system, similar to SCADA or MES systems in traditional manufacturing.

As part of a case-based design science approach, we have presented a functional scope and overall architecture for such a software manufacturing control system, and have described the design and prototype implementation of the artifact for the case of a specific software factory. This software manufacturing control system prototype does not provide fundamentally new information, but collects, aggregates and integrates information over time, across various repositories and build pipelines, and from different DevOps tools and technologies. Therefore, this control system does not provide new possibilities per se to optimize processes and improve output in software factories, as this can be done today by analyzing in detail the data produced by the various tools. However, aggregating and providing this information with short latency times, offers the opportunity to fundamentally reduce the lag times for such optimizations and improvements. Though the design as a search process is still ongoing, we have presented some use cases where the added value was validated in the case study. We have also discussed some preliminary findings and results of the implementation in the target software factory.

Investigating the creation of such a software manufacturing control system is believed to make some contributions. First, we have identified and validated a need for integrated control in today's state of the art automated DevOps environments. Second, we have designed an architecture that enables the rather straightforward creation of such integrated software manufacturing control systems in most contemporary software factories. Third, we have described and validated a number of detailed scenarios that can leverage such an integrated control system to improve the output of such software factories. Fourth, we have empirically shown that the implementation of such a control system can improve the consistency and structure across a state-of-the-art DevOps infrastructure.

Next to these contributions, it is clear that this investigation is also subject to a number of limitations. First, the case-based approach means that the integrated system has been created for a single software factory, though this factory does include for instance code generators. Second, the major part of the added value through optimizations and improvements, enabled by the drastic reduction of the lag times in the control processes, has yet to be confirmed empirically. However, its design has been validated by some key actors in our case study, and we have already verified empirically improvements in structure and consistency across the factory infrastructure, such as streamlining expansion settings and DevOps configurations across application projects.

REFERENCES

[1] H. Mannaert, K. De Cock, and J. Faes, "Exploring the creation and added value of manufacturing control systems for software factories," in Proceedings of the Eighteenth International Conference on Software Engineering Advances (ICSEA 2023), 2023, pp. 14–19.

[2] A. Marc, "Why Software Is Eating the World," URL: https://a16z.com/2011/08/20/why-software-is-eating-the-world/, 2011, [accessed: 2023-07-27].

[3] M. D. McIlroy, “Mass produced software components,” in Proceedings of NATO Software Engineering Conference, Garmisch, Germany, October 1968, pp. 138–155.

[4] S. E. Institute, “Software Product Lines Collection,” URL: https://resources.sei.cmu.edu/library/asset-view.cfm?assetid=513819, 2023, [accessed: 2023-07-27].

[5] C. Krueger, “Introduction to the emerging practice software product line development,” Methods and Tools, vol. 14, no. 3, 2006, pp. 3–15.

[6] J. Greenfield, K. Short, and S. Cook, Steve; Kent, Software Factories: Assembling Applications with Patterns, Models, Frameworks, and Tools. Wiley, 2004.

[7] T. Saeed, “Current issues in software re-usability: A critical review of the methodological & legal issues,” Journal of Software Engineering and Applications, vol. 13, no. 9, 2020, pp. 206–217.

[8] H. Mannaert, J. Verelst, and P. De Bruyn, Normalized Systems Theory: From Foundations for Evolvable Software Toward a General Theory for Evolvable Design. Koppa, 2016.

[9] H. Mannaert, K. De Cock, P. Uhnak, and J. Verelst, “On the realization of meta-circular code generation and two-sided collaborative metaprogramming,” International Journal on Advances in Software, no. 13, 2020, pp. 149–159.

[10] H. Mannaert, J. Verelst, and K. Ven, “The transformation of requirements into software primitives: Studying evolvability based on systems theoretic stability,” Science of Computer Programming, vol. 76, no. 12, 2011, pp. 1210–1222, special Issue on Software Evolution, Adaptability and Variability.

[11] M. Courtemanche, E. Mell, and A. S. Gills, “What Is DevOps? The Ultimate Guide,” URL: https://www.techtarget.com/searchitoperations/definition/DevOps, 2023, [accessed: 2023-07-27].

[12] R. T. Yarlagadda, “Devops and its practices,” International Journal of Creative Research Thoughts (IJCRT), vol. 9, no. 3, 2021, pp. 111–119.

[13] A. Kumar, M. Nadeem, and S. M., “Assessing the maturity of devops practices in software industry: An empirical study of helena2 dataset,” in Proceedings of the 26th International Conference on Evaluation and Assessment in Software Engineering (EASE 22), 2022, pp. 428–432.

[14] S. M. Faaiz, S. U. R. Khan, S. Hussain, W. Wang, and N. Ibrahim, “A study on management challenges and practices in devops,” in Proceedings of the 27th International Conference on Evaluation and Assessment in Software Engineering (EASE 23), 2023, pp. 430–437.

[15] M. Bhat, “How to Select DevSecOps Tools for Secure Software Delivery,” URL: https://www.gartner.com/en/documents/4131199, 2023, [accessed: 2023-10-10].

[16] Dynatrace, “Observability and security convergence: Enabling faster, more secure innovation in the cloud,” URL: https://assets.dynatrace.com/en/docs/report/bae1393-rp-2023-global-cio-report-observability-security-convergence.pdf, 2023, [accessed: 2023-10-10].

[17] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design science in information systems research,” MIS Quarterly, vol. 28, no. 1, 2004, pp. 75–105.

[18] H. Mannaert, T. Van Waes, and F. Hannes, “Toward a rejuvenation factory for software landscapes,” in Proceedings of the Sixteenth International Conference on Pervasive Patterns and Applications (PATTERNS 2024), 2024, pp. 13–18.

[19] R. Sturn, “Division of labor: History of the concept,” in International Encyclopedia of the Social & Behavioral Sciences. Oxford: Elsevier, 2015, pp. 601–605.

[20] H. Mannaert, J. Verelst, and K. Ven, “Towards evolvable software architectures based on systems theoretic stability,” Software: Practice and Experience, vol. 42, no. 1, 2012, pp. 89–116.

[21] N. Telecommunications and I. A. (NTIA), “Software Bill of Materials,” URL: https://www.ntia.gov/page/software-bill-materials, 2021, [accessed: 2024-03-05].

Regional Feature Importance for Error Analysis in Manufacturing

Importance Measure that Reveal Insights

Valentin Göttisheim, Holger Ziekow,
Peter Schanbacher
Furtwangen University
78120 Furtwangen, Germany
{valentin.goettisheim, holger.ziekow,
peter.schanbacher}@hs-furtwangen.de

Djafar Ould-Abdelsam
Université de Haute-Alsace
IRIMAS Laboratory, Université de Haute-Alsace
68100 Mulhouse, France
djaffar.ould-abdeslam@uha.fr

Abstract - **Quality management in manufacturing can benefit from integration of artificial intelligence to detect and analyze errors in production. However, finding the causes of errors requires not only accurate predictions, but also suitable explanations of the underlying data analysis processes. This paper extends our previous work on a novel approach to measure the importance of features for error analysis. This approach bridges the gap between global and local importance and introduces the concept of regional feature importance, which captures the impact of features in specific regions of the feature space, rather than globally or locally. We generalize this method as a task of partitioning the feature space and aggregating the local importance of features within these regions. Our findings demonstrate that this approach can reveal interesting and actionable insights for quality management in manufacturing.**

Keywords - eXplainable Artificial Intelligence; XAI; SHapley Additive exPlanations; SHAP; feature importance, manufacturing quality management; error analysis.

I. INTRODUCTION

In this paper, we expand our earlier work [1] on regional feature importance (RFI) measures for quality management in manufacturing. This work generalizes the concept of RFI as a two-step process of partitioning and aggregation. The partitioning step divides the feature space into regions based on certain criteria, and the aggregation step combines the feature importance values within each region. Additionally, we extend our previous work by introducing new measures and demonstrate the usefulness of the proposed RFIs on synthetic data and a real-world dataset.

Feature importance measures are valuable tools for analyzing complex, high-dimensional production data to identify the causes of errors or defects. Quality management in modern manufacturing processes involves extensive testing and collection of detailed measurements along production lines. However, quality managers often struggle to pinpoint error causes within large data sets [2]. Artificial Intelligence (AI), combined with eXplainable AI (XAI), can utilize such data to predict errors [3] and provide insights to quality engineers by identifying potential causes of production errors [4]. Feature importance metrics can reveal these features that constitute potential error causes and reveal interesting insights for quality managers. However, existing feature importance measures are not tailored to this task [4] [5].

Our work is rooted in a research project with a German manufacturer [6]. Here, combining human expertise with AI-based data analysis is desirable for error analysis in production lines. This is because (a) quality managers seek to understand the error causes and may not blindly trust AI-based results, and (b) human experts have background knowledge and a deep understanding of the production process that the AI does not have access to. Hence, this work explicitly involves human experts in the loop and focuses on using AI models for providing input to human analysts.

This work targets typical manufacturing setups, where production lines comprise a sequence of production steps and several test stations along the production line. Test stations perform measurements on each product at different steps of the production. This leads to detailed records of individual product instances that can include hundreds of thousands of measurements per product [7]. However, the high number of different measurements poses challenges for finding causes of errors in the data. Moreover, errors are rare in modern manufacturing processes which usually are highly optimized [8]. Quality management is often about driving down rare – but still costly – errors. Yet, existing applications have successfully used such high-dimensional to build AI models for predicting production errors [7][9]. The aim of such models is to take measurements from test stations early in the production sequence and predict errors that occur downstream in the production line. If errors can be predicted early with sufficient reliability, products can be removed early in the process, and costs for downstream production steps can be avoided [2].

Furthermore, such AI models can be analyzed to hint at the cause of errors. We leverage this capability to provide insights to human experts in quality management. Existing works use feature importance measures to identify quality measurements that are relevant in predicting and explaining errors. For example, if a heat measurement of an oven is important in predicting errors, then errors may be avoided by adjusting the temperature setting. Identifying such interesting measurements among the thousands of data points can help quality managers to find error causes and improve production [9]. However, existing importance measures are not tailored

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

to find features that are interesting for inspection in error cause analysis. Instead, they take a global view and capture how much a model relies on a given feature on average. As we demonstrate in this paper, such a global view often fails when it comes to spotting rare but strong relations that lead to actionable insight in error analysis.

In contrast to global importance measures, XAI methods like Shapley Additive Explanations (SHAP) [11] and LIME [12] provide local explanations for the impact of features on a prediction. These methods estimate the impact of features on individual data instances. However, analyzing a single data instance in isolation may not yield enough context to draw actionable conclusions and, thereby, limiting the usefulness in quality management.

Our work introduces feature importance measures that bridges the gap between global and local feature importance. We refer to this as regional feature importance (RFI). Building on this concept, we extend and refine our previous paper [1]. That is, we analyze sets of local feature importance values for interesting effects. The result of this analysis is captured in new importance measures that capture different interesting aspects. In this paper, we mathematically define our applied notion of interestingness. Intuitively, we consider a feature interesting if it hints at actionable insight for quality managers. Such actions include setting thresholds in quality checks or adjusting processes to avoid specific value ranges. Intuitively, drastic changes in error rates and high error rates in well-defined parts of a value range make features interesting. This paper presents importance measures that formalize these concepts of interestingness and translate them into an importance score. Specifically in this work, we make the following key contributions:

1) Extending and formally defining novel feature importance measures that are tailored to finding relevant features for quality management in manufacturing,
2) proposing a two-step process of partitioning and aggregating to construct the RFI measures,
3) providing five metrics to determine the RFI measures based on different criteria, and
4) evaluating the proposed measures on real-world data and comparing them with established importance measures.

With these contributions, we aim to assist human experts in quality management to better leverage results from AI models for driving their analysis.

The remainder of this paper is structured as follows: In Section II, we briefly summarize the corresponding background. In Section III, approaches to derive the regional feature importance are proposed which are illustrated on synthetic data in Section IV, and evaluated on a real-world dataset in Section V. In Section VI, we discuss related work and conclude in Section VII.

II. BACKGROUND

When using Machine Learning (ML) support for error analysis in quality management processes, feature importance metrics can become a tool to rank and identify features that are suitable to guide Quality Engineers (QE) in finding error causes in production. Such a process inspired the present work is carried out in the production of an industry partner in the research project [5]. ML-driven quality management processes here focus on QEs as primary actors. Using ML support, QEs are intended to analyze production and take corrective maintenance steps in production. However, the development and deployment of models for the ML support system are embedded in automated pipelines and maintained by data scientists. The automated ML pipeline includes several steps like data preprocessing, i.e., feature selection or evaluation of model performances through cost-sensitive metrics [3]. As such, the system is designed to enable QEs to use ML support for error causes analysis, but not to engage with the technical depth of the ML system.

A reference process focusing on QEs intended to investigate errors in production is laid out in [2]. Key steps include the selection of production data for the automated ML pipeline. Later steps involve error identification and correction in production using ML support. To identify error causes the QE is intended to use feature importance to find features that suits as explanations for error causes.

SHAP is one of the more recent advancements in the field of XAI, focusing on the interpretability of ML models. SHAP targets instance-based, as opposed to global, model explanation. By aggregating explanations of instances, it is possible to evaluate the importance of features incorporating aspects of interest to guide QEs in error cause analysis [5]. SHAP evaluates the marginal contribution a feature has on its model output. The contribution $\phi_f \in \mathbb{R}$ for a feature f with model m is attributed using Shapley Values from game theory:

$$\phi_f = \sum_{S \subseteq N \setminus \{f\}} \frac{|S|!\,(M - |S| - 1)!}{M!} [m_x(S \cup \{f\}) - m_x(S)],$$

where M is the number of all features, S is the set of input values, and |S| is the magnitude of S (for example, $S = x_1, x_2, \ldots, x_{f-1}, x_{f+1}, x_n$ and $|S| = n - 1$). One way to compute the feature contributions is an explanation model $e(z') = \phi_0 + \sum_{f=1}^{M} \phi_f z'_f$ where $z' \in \{0,1\}^M$[11]. This is the weighted average over all feature contributions. The explanation model is computed using the mapping $m_x(S) = m(e_x(z'))$ which maps all input values S to whether the feature is being used ($z' = 1$) or not known ($z' = 0$). However, SHAP values can also be efficient computed using model specific methods such as for tree-based model exploiting the internal structure of the model [13] .

SHAP decomposes the feature contribution into main effect and interaction effect with other features [13]. The main effect of a feature is the average marginal contribution of that feature across all possible coalitions of other features. The interaction effect of a feature with another feature is the difference between the joint contribution of both features and the sum of their individual contributions. The advantage of distinguishing the main and interaction effects is that it can reveal how features influence the model output not only by themselves, but also by interacting with other features. This can help to understand the complex and nonlinear relationships between features and the model output. Therefore, SHAP values can be decomposed in main and interaction effect by:

$$\phi_f(x_i) = \phi_{f,f}(x_i) + \sum_{j \neq f} \phi_{f,j}(x_i)$$

Where $\phi_{f,j}(x_i)$ is the SHAP interaction value between a feature f and a feature j on instance x_i. Moreover, $\phi_{f,f}(x_i)$ then is the main effect which is the resulting effects of feature f with itself.

In the following section we propose SHAP-based importance measures that are tailored to the task of quality management in manufacturing.

III. REGIONAL FEATURE IMPORTANCE

RFI is an approach to measure the contribution of a feature to the prediction of a model in a specific region of the feature space [5]. It bridges the gap between global and local feature importance [14][15]. This is unlike global importance, which assesses features over the entire dataset, and local importance, which focuses on the individual prediction [16]. Regional importance focuses on specific regions in the data and aims to identify those that are most relevant for explaining the error cause in manufacturing processes.

We propose a two-step process of partitioning and aggregation to construct the RFI. Partitioning are methods for dividing the feature value range into clusters that capture the local patterns and behaviors of the features. The partitioning is based on criteria, such as distance of data points, output predictions or intervals of feature values. This step isolates distinct areas where feature behavior is expected to be similar and allows interpretations about different contexts and regions. Following the partitioning, SHAP values within the region are aggregated to determine an importance score. These aggregations are based on criteria such as mean SHAP values, error frequency or change in SHAP values to pinpoint interesting prediction contributions of features.

To describe the RFI we define $X = \{x_1, x_2, \dots, x_n\}$ as the set of data point in the dataset X. However, for simplicity we also refer to data points as instance $x \in X$. Correspondingly, we define the set of labels $Y = \{y_1, y_2, \dots, y_n\}$ as the real data labels. We denote M as the machine learning model that maps instance x to a prediction $M(x) = \hat{y}$. We use SHAP values $\phi_f(x)$ as a measure to quantify the contribution of an instance x to the prediction for a feature f. We seek to evaluate a scoring-function $g: g(f, X, y, M) \to \mathbb{R}$ that aggregates SHAP values and scores "interesting" features high.

In the following sections, we provide the two-step process of partitioning and aggregations to construct the RFI in detail. Moreover, we propose several methods for partitioning and aggregations, provide mathematical formulations and intuition for interpretation and reasoning.

A. *Partitionings*

To determine the RFI, first the feature value range is partitioned into clusters that optimize the feature importance score:

$$g(f, X, y, M) = \max_{C \in P(f)} A(f, X, C)$$

Here, $A(f, X, C)$ represents the aggregations later defined, which are obtained for sets of clusters $C \in P(f)$. $P(f)$ represents the proposed partitioning functions, resulting in sets of clusters $P(f)$={$C_1, C_2, \dots C_m$} for feature f. Each cluster $c \in C_f$ is a disjunct subset of the data X such that $c \subseteq X$. In the following, we describe the used partitioning methods $P(f)$ to obtain C_f in detail.

Decision-Tree Partitioning: This method divides the feature value range into segments based on the splits of a decision tree that is trained on the feature values f and the output labels y. Each split is determined by selecting a threshold that maximize the reduction in impurity only considering one feature f. As impurity measure, we used the Gini $1 - \sum_{i=1}^{k} p_i^2$, where p_i is the proportion of data points in class i at a given node. Clusters C are then determined by the node in which data points x falls. The intuition is that features that have interesting patterns of SHAP values in regions that are predictive of the output labels are more indicative of error causes. This method can handle complex feature value distributions, as it creates partitions that are based on the feature value patterns and the output labels y regardless of the feature value range.

K-means Partitioning: This method uses the k-means algorithm to segment the feature values f into clusters, aiming to reduce the within-cluster variance. It minimizes the within-cluster sum (WCSS) of squares, defined as WCSS = $\sum_{x_f \in c_j} \left|\left| x_f - \mu_j \right|\right|^2$ where μ_j is the centroid of cluster c_j. This approach groups feature values close to a segment's centroid and separates those further away, thereby identifying areas with consistent feature values and therefore, sheds light on the local structure of the data, providing valuable insights into feature behavior.

Interval-based Partitioning: This method linearly divides the feature range into equal-sized intervals. This divides the entire range of f into partitions, into intervals I_j covering an equal portion of the feature's value range. Every data point x is allocated to the cluster c_j based on the intervals I_j range in which a feature value x_f lies. This method does not account for density or distribution patterns, instead, it examines how the characteristics changes across the features spectrum, and therefore, exhibits patterns which changes across the feature value range.

Hierarchical-based Partitioning: This method partitions the feature based on hierarchical clustering. Clusters are formed based on the pairwise Euclidean distances between data points within the feature f. To obtain the clusters the resulting dendrogram is cut at a specific level or the number of partitions is specified. Each cluster $c_i \in C$ comprises data points that are closely related in terms of their feature value. Therefore, this partitioning emphasizes similarity in the feature value.

With the discussed partitioning approaches, we laid out diverse approaches to segment a feature space. Each method offers a unique insight, emphasis different aspects of data relationships and distribution patterns. We now discuss the aggregations, facilitating the interpretation and synthesis of findings to gather actionable insights from the RFI.

B. Aggregations

After partitioning, we now describe the aggregations as second step to construct the RFI score $g(f, X, y, M) = \max_{C \in P(f)} A\,(f, X, C)$ where A denotes the aggregation method applied to each cluster C that has been derived from a partitioning method $P(f)$ of feature f.

Mean-Shap: The aggregation $A_{\text{mean}}(f, X, C)$ quantifies the average SHAP value across the data points of each cluster C. Formally, it is defined as:

$$A_{\text{mean}}(f, X, C) = \frac{1}{|C|} \sum_{x \in C} \phi_f(x)$$

Here, $|C|$is the number of data points in cluster C, and $\phi_f(x)$ is the SHAP value of feature f for a data point x. This aggregation method provides an average measure of the influence of a feature within a specific cluster, giving an overall indication of its importance within its feature value range.

Main-SHAP: Main SHAP quantifies the average main effect within a specific cluster, isolating these effects from interactions with other features. This metric measures the effect caused by the feature itself rather than of interactions. The intuition is that it provides a quantification of the contribution from the intrinsic influence of the feature on error in production, offering a possibly simple to interpret explanation in the domain context. This aggregation method is crucial for pinpointing features as explanation, facilitating which features are fundamental. The Main-SHAP is defined using the main SHAP values $\phi_{f,f}(x)$ described in the background section and formally expressed as:

$$A_{Main}\,(f, X, C) = \frac{1}{|C|} \sum_{x \in C} \phi_{f,f}(x)$$

Error-Shap: This aggregation method sums up the SHAP values of instances indicating errors or faulty products in the data. The intuition is that features that have high SHAP values for the errors are more relevant for explaining the causes. Formally, let x_e be the subset of data points $x_e \subseteq X$ where the actual labels indicating errors in the products. Error-Shap is then defined as:

$$A_{\text{err}}(f, X, C) = \sum_{x_e \in C} \phi_f(x_e)$$

Here, $\phi_f(x_e)$ represents the SHAP value for the error instance x_e. This aggregation sums up the SHAP values for across all error instances, emphasizing the relevance of the feature in explaining the causes of errors.

Error-Rate-Shap: This method multiplies the shap value of a feature by the error rate of the corresponding feature value. The intuition is that features that have high SHAP values and high error rates are more interesting for finding error causes. Formally, Error-Rate-Shap is defined as:

$$A_{\text{erate}}(f, X, C) = \sum_{x \in C} \phi_f(x) \cdot e(C)$$

where $e(C)$ represents the error rate within cluster $C$and is calculated as $|x_e|/|x_{ne}|$ within the same cluster. Specifically, as the proportion of the amount of error instances $|x_e|$ compared to the amount of non-error instances $|x_{ne}|$ for x_{ne} as subset of data points $x_{ne} \subseteq X$ where the actual labels indicate a non-error label. This error ratio $e(C)$ quantifies how often that cluster is associated with errors.

Slope-Shap: This measure assesses how rapidly the SHAP values of a feature change over its value range. It aims to identify features with significant SHAP value shifts, indicative of potential error causes. After partitioning, a rolling window method is applied to calculate the slope of the mean SHAP values across the window $w \in W$. The Slope-Shap aggregation is then defined as the absolute sum of these slopes within each window.

$$A_{\text{slope}}(f, X, C) = \sum_{w \in W} \left| \frac{\overline{\phi}_f(X_{w+1}) - \overline{\phi}_f(X_w)}{\Delta} \right|$$

Here, W denotes the set of rolling windows, X_w is the subset of X for the w-th window, and Δ represents the width of each window. $\overline{\phi_f}$ is denoted as the mean SHAP value of X_f, the set of all data points where feature f is present, and calculated as $\overline{\phi}_f(X') = \frac{1}{|X'|} \sum_{x \in X'} \phi_f(x)$ with $|X'|$ representing the number of data points of X'.

Z-score SHAP: This aggregation quantifies the deviation of SHAP values within a specific region from the average contribution of a feature. It provides a measure of how abnormal extreme feature contributions are in relation to the features typical impact. A high absolute Z-score indicates a significant deviation, pointing to features that have an unusually strong impact on predictions within regions compared to their average impact across all data. The Z-score SHAP is defined as follows:

$$A_{zscore}(f, X, C) = \frac{\sum_{x \in C} \phi_f(x) - \mu_f}{\sigma_f}$$

where μ_f is the mean SHAP value and σ_f is the standard deviation of the SHAP values of feature f across the entire dataset.

In this section, we have presented the concept of RFI and proposed several partitioning and aggregation methods to capture interesting and relevant aspects of features for error analysis in manufacturing. To illustrate the usefulness and effectiveness of our proposed RFI measures, we now apply them to synthetic data scenarios that mimic common error situations in manufacturing processes.

IV. SYNTHETIC DATA EVALUATION

In the following, we illustrate the proposed regional feature importance using synthetic data, specifically designed to reflect characteristics of manufacturing situations, focusing on identifying features that are most interesting for explaining error causes. We focused on two specific error scenarios (A) Tail Error, and (B) Segment Error. Each scenario was designed to mimic distinct types of errors encountered in manufacturing data.

A) Tail Error: Involves a higher error rate in the tail of a normal distribution, representing errors that are rare but relevant.

B) Segment Error: This scenario focuses on a feature that strongly impacts the label, but only within a small range.

Global feature importance (GFI) measures aggregate the importance of features across the entire data distribution. This aggregation dilutes the impact of features for error analysis that focuses on rare events which may only occur in certain regions of the feature value range. The RFI highlights these critical regions, while global importance may overlook them due to averaging effects, leading to a misrepresentation of the feature's true impact in error analysis.

The use of synthetic data has the advantage that the ground truth is known. Here the correctness can be assessed by determining if the importance metric correctly identifies the most relevant feature (i.e., pinpointing relevant features as explanations for error analysis). Each synthetic data scenario consists of three main features: Target, Trap, and Noise. The "Target" feature is directly tied to error rates, "Trap" is a deceptive feature that ranks high in global importance but is less interesting in error analysis, and "Noise" introduces a minimal error rate, serving as a control variable. Each scenario is modeled as a binary classification task with 1 labeled as an error instance indicating a faulty product in production or 0 if not. We employ XGBoost [17] as a predictive model trained on 10,000 data points on which it achieves a perfect training ROC AUC score of 1. Note, that potential overfitting is not a concern for our experiments as they target feature importance and not prediction quality. Subsequently, global importance measures, such as Weight, Gain, Cover, or Abs. Mean SHAP and the RFI is computed. This process is repeated 10 times. The findings are presented in Table 1 and Table 2 as mean importance scores alongside the standard deviations.

A. Tail Error

This scenario simulates errors occurring in the tail regions of data distributions, often representing rare yet critical error cases. These errors are challenging to detect because they occur infrequently, yet they can have a considerable impact on the overall results in manufacturing. The scenario comprises three features "Target", "Noise", "Trap", and the binary label y that is affected by the error characteristics. We introduced anomalies specifically in the tail of the Target feature, ensuring these errors are clearly noticeable but overall not frequent.

- **Target:** A normally distributed feature that causes a 10% error rate based on a 0.02 quantile in the upper tail of its value range.
- **Trap:** A normally distributed feature that causes a 4% error rate over 0.5 quantiles of its range.
- **Noise**: A normally distributed feature that introduces a small amount (1%) of random noise in the label.

We argue that feature Target, despite its infrequent yet strong relations to errors, presents a more interesting case for error analysis than the Trap feature. Although the Trap feature encompasses more errors, it's less interesting as it comprises a much broader value range.

The global importance measures, as reported in Table 1, show that the Trap feature ranks highest, while the Target

TABLE 1. "EVALUATION RESULT GFI": RANKING THE DECEPTIVE FEATURE "TRAP" IMPORTANT

GFI	**Rank**		
	1	*2*	*3*
Cover	Trap 122.94/7.2	Target 83.76/7.22	Noise 75.28/3.10
Gain	Trap 1.49/0.05	Target 1.33/0.05	Noise 1.32/0.06
Mean Abs. Shap	Trap 0.52/0.05	Target 0.26/0.02	Noise 0.24/0.04
Total Cover	Trap 77331.80/5177.75	Target 52595.52/3989.74	Noise 46620.20/3220.44
Total Gain	Trap 936.7/34.4	Target 839.8/54.5	Noise 816.0/58.7
Weight	Trap 623.1/34.9	Target 619.0/30.0	Noise 607.9/24.7

Notation: feature (mean/std)

TABLE 2. " EVALUATION RESULT RFI": RANKING THE INTERESTING FEATURE "TARGET" AS IMPORTANT

RFI		Rank		
A	*P*	*1*	*2*	*3*
Mean SHAP	Tree	Target 1.34/0.13	Trap 1.09/0.15	Noise 0.5/0.11
	Hierarch.	Target 2.37/0.61	Trap 0.88/0.15	Noise 0.53/0.34
	Interval	Target 1.07/0.14	Trap 1.01/0.13	Noise 0.4/0.11
	K-means	Target 2.05/0.42	Trap 0.93/0.12	Noise 0.42/0.2
Error Rate SHAP	Tree	Target 0.18/0.03	Trap 0.1/0.02	Noise 0.03/0.01
	Hierarch.	Target 0.76/0.35	Trap 0.07/0.07	Noise 0.07/0.14
	Interval	Target 0.11/0.02	Trap 0.08/0.02	Noise 0.02/0.01
	K-means	Target 0.56/0.28	Trap 0.06/0.01	Noise 0.02/0.01
Error SHAP	Tree	Target 0.16/0.02	Trap 0.14/0.02	Noise 0.04/0.01
	Hierarch.	Target 0.13/0.03	Trap 0.1/0.01	Noise 0.03/0.01
	Interval	Target 0.13/0.04	Trap 0.1/0.02	Noise 0.03/0.01
	K-means	Target 0.13/0.03	Trap 0.1/0.02	Noise 0.03/0.01
Main SHAP	Tree	Target 1.29/0.12	Trap 1.15/0.18	Noise 0.53/0.11
	Hierarch.	Target 1.92/0.37	Trap 0.82/0.18	Noise 0.74/0.45
	Interval	Target 1.0/0.09	Trap 0.98/0.15	Noise 0.41/0.11
	K-means	Target 1.69/0.3	Trap 0.91/0.13	Noise 0.72/0.34

Notation: feature (mean/std)

feature, which we consider most important, ranks second. As anticipated, feature Noise, which was intended to be the least important, ranks last. These results show that global feature importance assigns higher importance to features that are less relevant for error analysis, thereby illustrating the limitations of global metrics for error analysis. The RFI aims to overcome the limitations of global measures by focusing on specific regions. The RFI is computed for every feature f using the proposed partitioning methods P to determine clusters $C \in P(f)$**.** The importance score $g(f, X, y, M)$ is the maximum value of a proposed aggregations A among all clusters C: $\max_{C \in P(f)} A(f, X, C)$.

In this scenario, Mean SHAP, Error Rate and Error SHAP are crucial metrics. The RFI importance scores g are presented in Table 2 as importance rank with the average importance scores g and its standard deviation over the 10 repetitions. Over all metrics except the Error SHAP with hierarchical partitioning, the feature Target is consistently ranked as the most important.

The RFI can also be illustrated as a scatter plot with curves the aggregations of SHAP values across the identified clusters as in Figure 1 shown. The blue points symbolize the SHAP values of individual data instances. The red points highlight the instances associated with errors in products. All plots show increased RFI metric scores g on the interval [2,4] indicating an increased importance of the region within the feature space that is relevant for explaining error causes.

This visual representation, along with Table 2, aids in understanding the connections between features and the potential error, i.e., the error in the tail regions of the data distribution. In the following, we discuss how these aggregations reveal interesting regions.

Mean SHAP: This aggregation calculates the average SHAP value for a cluster across all instances of that cluster. SHAP values quantify the contribution of a feature to the prediction for an instance. Therefore, the Mean SHAP

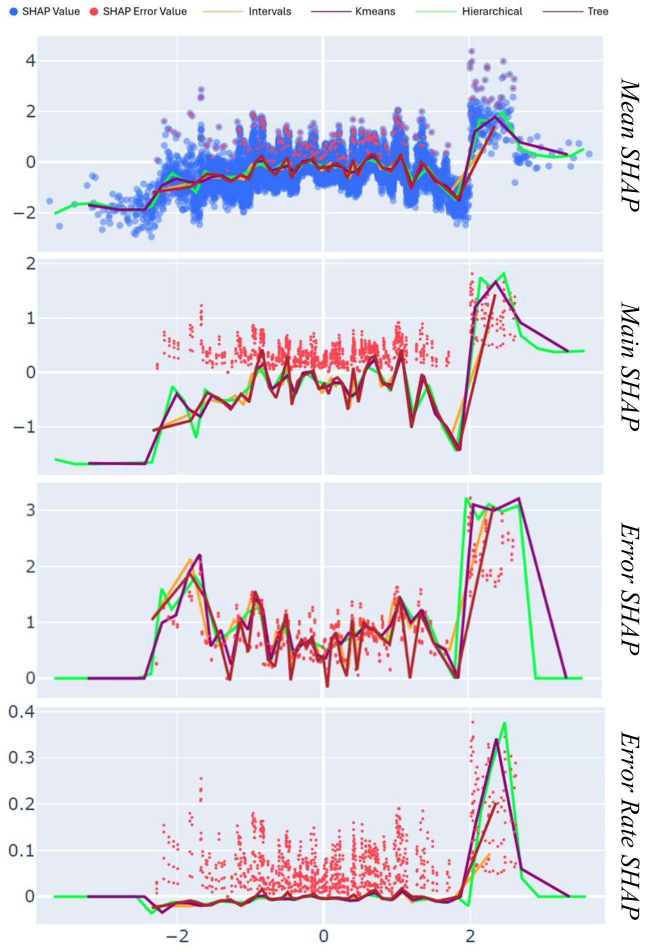


Figure 1. "Evaluation RFI Target feature" RFI aggregations over all partitions of the Target feature showing increased values for features values greater than two.

aggregation provides a measure of the average impact of the cluster on error predictions, highlighting clusters that have a strong relation to errors. Figure 1 Mean SHAP shows the increased average impact for a feature value greater than two.

Main SHAP: Main SHAP quantifies the average main effect within a specific cluster, isolating the effects from interactions with other features. This metric shed light on the standalone contribution of features to the model's output, revealing the intrinsic influence of the feature. This aggregation method is crucial for pinpointing features as explanations, facilitating which features are fundamental to predictions. The Main SHAP metric is instrumental in identifying key drivers by focusing on the direct impact of individual features. Figure 1 shows the Main SHAP plot.

Error Rate SHAP: Considers the error rate alongside with the region's impact. It is computed by multiplying the cluster mean SHAP with the error rate of the cluster. The error rate is the proportion of instances associated with errors relative to those that are not. This aggregation highlights impactful regions with frequent errors.

Error SHAP: This measure considerers SHAP values of all instances labeled as errors in a region. It offers a direct insight into which regions are responsible for errors, regardless of how often those errors occur. This highlights significant errors even in regions without strong associations with errors. SHAP plots for Error and Error Rate SHAP are also presented in Figure 1.

B. *Segment-Error*

The second scenario resembles the challenge of identifying production errors in a small value range, targeting production problem that affects the overall production process performance. The scenario follows the setup described earlier. However, the following features are considered as training data:

- **Target**: A uniformly distributed feature that causes a higher error rate of 10% in a small segment of its range.
- **Trap**: A uniformly distributed feature that causes a moderate error rate in five segments of its range in varied intensity of 0.2-1 % error rate.
- **Noise**: A uniformly continuous variable that accounts for 1% errors of the data as noise.

We argue that feature Target, as of its localized yet significant relations to errors, presents a more interesting case for error analysis than the Trap feature. Although the Trap feature encompasses more errors, it's less interesting as it again comprises a broader value range. The global importance measures, as reported in Table 3, show that the Trap feature ranks highest, while the Target feature, which we consider most important, ranks second. As anticipated, feature Noise, which was intended to be the least important, ranks last. These results show that global feature importance assigns higher importance to features that are less interesting for error analysis, thereby illustrating the limitations of global metrics for error analysis.

TABLE 3. " EVALUATION RESULT GFI" RANKING THE DECEPTIVE FEATURE "TRAP" AS IMPORTANT

GFI	Rank		
	1	*2*	*3*
Cover	Trap 144.75/12.76	Target 116.39/8.49	Noise 105.70/9.51
Gain	Trap 1.64/0.07	Target 1.59/0.06	Noise 1.53/0.06
Mean Abs. Shap	Trap 0.31/0.02	Target 0.22/0.02	Noise 0.16/0.02
Total Cover	Trap 93895.74/7866.96	Target 77933.30/7179.09	Noise 67254.76/6711.44
Total Gain	Trap 1066.32/41.3	Target 1063.8/54.0	Noise 975.96/59.6
Weight	Target 669.90/44.50	Trap 649.10/18.3	Noise 636.50/35.3

Notation: feature (mean/std)

Subsequently, the RFI is computed for every feature f using the proposed partitioning methods P to determine clusters $C \in P(f)$. The importance score $g(f, X, y, M)$ is than examined as the maximum value among all clusters $C: \max_{C \in P(f)} A\,(f, X, C)$. This illustration focuses on the proposed aggregations Slope SHAP and Z-Score SHAP for which the results of g in Table 3 are reported based on 10 repetitions, and the scatter plots with its RFI curves in Figure 2.

The RFI results in Table 4 shows that both metrics rank the more interesting feature Target as more important. The scatter plots in Figure 2 show the increased mean SHAP

TABLE 4. " EVALUATION RESULT RFI": RANKING THE INTERESTING FEATURE "TARGET" AS IMPORTANT

RFI		Rank		
A	*P*	*1*	*2*	*3*
Slope SHAP	Tree	Target 0.46/0.08	Trap 0.29/0.07	Noise 0.25/0.06
	Hierarch.	Target 0.43/0.05	Trap 0.27/0.07	Noise 0.23/0.05
	Interval	Target 0.44/0.05	Trap 0.27/0.07	Noise 0.23/0.06
	K-means	Target 0.42/0.06	Trap 0.27/0.08	Noise 0.22/0.06
Z-Score SHAP	Tree	Target 559/119.3	Trap 421/66.63	Noise 284/41.46
	Hierarch.	Target 486/73.44	Trap 377/52.87	Noise 245/59.44
	Interval	Target 470/34.6	Trap 348/40.74	Noise 230/26.65
	K-means	Target 479/59.99	Trap 351/40.95	Noise 233/42.2

Notation: feature (mean/std)

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

values in the interval [0.4,0.425]. In the following, we discuss how the other both aggregations reveal interesting regions.

Z-Score SHAP: The Z-Score SHAP metric quantifies the deviation of SHAP values within a specific region from the average contribution of a feature. It is measured in standard deviations from the mean impact of a feature across the entire dataset. It provides a measure of how anomalously a feature behaves in relation to its general impact. A high absolute Z-score indicates a significant deviation, pointing to features that have an unusually strong impact on predictions within regions compared to their average impact across all data. In this illustration, this can be observed over the interval [0.4,0.425] in the scatter plots in Figure 2.

Slope-SHAP: This metric is used to identifying points or regions where the contribution of the feature changes significantly. Slope SHAP quantifies the rate of change in the mean SHAP values by calculating the slope of these values using a rolling window over the partitioned feature value range. This aggregation considers the absolute values, thereby capturing the overall variability in the feature's impact. High Slope-SHAP values indicate features that exhibit pronounced shifts in their influence at specific thresholds, signaling potential transition points that could lead to errors in production. This method enables the pinpointing of errors that are caused by exceeding a threshold, which can be mitigated by setting or adjusting an alarm threshold in production. In this illustration, this can be observed over the intervals [0.3,0.4] and [0.4,0.5] in the scatter plots in Figure 2.

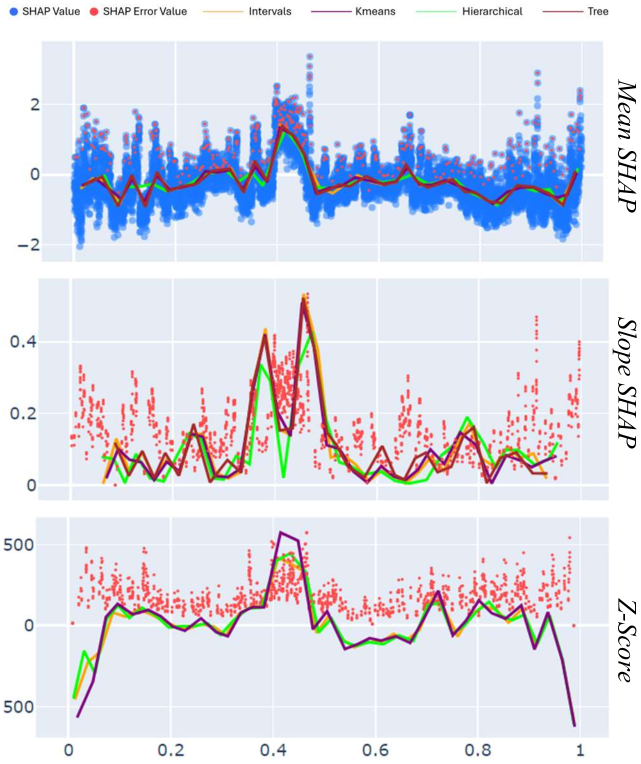


Figure 2. "Evaluation RFI Target feature" RFI aggregations over all partitions of the Target feature showing increased values for features values between 0.4 and 0.45.

These illustrations shows that the regional feature importance extend the insights gained from global feature importance measures in identifying the features that are most relevant for explaining the causes of errors in the synthetic data. The regional feature importance measures can capture the features that are more interesting for error analysis, while the global feature importance measures can be misleading or insufficient, as they take a global view and ignore rare but strong relations.

V. REAL WORLD DATA EXPERIMENT

In this section, we evaluate the RFI measures on a real-world data set form the steel manufacturing domain [18]. The dataset contains 27 features and 52407 instances related to the quality of steel plates, and 158 binary labels indicating a product error, i.e., a production fault. We trained an XGBoost model which achieves a ROC AUC of 1 on the training data. Again, potential overfitting is not a concern for our experiments as they target feature importance and not prediction quality. This high performance ensures that the model can effectively capture the relationships within the data, making it suitable for evaluating the feature importance measures.

The RFIs are computed as each feature is divided in 20 partitions resulting in different cluster sizes depending on the feature. To compute the Slope SHAP aggregation a window size of five is used. In the following discussion, we compare the global and the regional feature importance rankings alongside with the RFI scatter plots.

The scatter plots show the curves of partitioning methods, where each break represents the mean of the cluster location over the feature value. For each cluster, the aggregations method is annotated on the right. As references the SHAP values of instances for error labels (red dots) are highlighted. However, only the Mean SHAP plot shows the real value of the error SHAP values. For all other scatter plots, the value is linearly scaled. Therefore, they show the right amplitude and position relative to each other, however, not the resulting SHAP values. These visualizations help in understanding how feature importance varies across different regions of the feature space, providing a nuanced view of feature contributions.

To identify the features that are most interesting for explaining the error causes and providing insights for quality management, we discuss the GFI and RFI of the top five rankings. The rankings are presented in the Table 5. The results show commonalities; however, there are also disagreements, which form the basis for the following discussion. Our analysis highlights the strengths and weaknesses of each method and provides a comprehensive understanding of feature importance in the context of manufacturing errors.

TABLE 5. "TOP FEATURE RANKINGS": STEEL DATA TOP 5 RANKINGS OF RFI AND GFI– HIGHLIGHTED TOP FIVE RANKINGS.

Aggregration *A*	**Partitions *P***	**Orientation_Index**	**Length_of_Conveyer**	**Edges_Y_Index**	**Outside_X_Index**	**Empty_Index**	**Minimum_of_Luminosity**	**Log_X_Index**	**Steel_Plate_Thickness**	**Y_Minimum**	**X_Minimum**	**Edges_Index**	**Luminosity_Index**	**Maximum_of_Luminosity**	**Square_Index**	**X_Maximum**	**X_Perimeter**	
Z-Score	Tree	**4**	6	21	**3**	13	18	20	9	**1**	8	17	11	12	10	**5**	**2**	*RFI*
	Hierarch.	**2**	7	17	23	11	9	15	**4**	**1**	6	13	12	**5**	10	**3**	22	
	Interval	**3**	**5**	22	**4**	13	19	20	14	**1**	6	12	9	11	10	8	**2**	
	K-means	**2**	9	16	21	8	11	14	**4**	**1**	**3**	12	13	**5**	7	10	23	
Error Rate	Tree	**1**	**2**	14	**5**	8	**3**	17	6	10	7	12	13	15	**4**	9	16	
	Hierarch.	**2**	**5**	13	24	8	**1**	15	**3**	**4**	6	12	16	10	7	9	23	
	Interval	**1**	**2**	14	6	7	**3**	17	12	11	**5**	9	13	16	**4**	8	15	
	K-means	**1**	**4**	11	24	9	6	14	**3**	**2**	**5**	12	16	10	7	8	23	
Error SHAP	Tree	**1**	**2**	14	7	**5**	**3**	19	10	**4**	13	8	12	6	9	17	16	
	Hierarch.	**1**	**2**	11	15	7	**3**	17	6	**4**	10	8	13	**5**	9	14	21	
	Interval	**1**	**3**	14	**4**	7	**2**	19	10	6	12	8	11	**5**	9	18	17	
	K-means	**1**	**3**	11	14	**5**	**2**	17	7	**4**	10	8	13	6	9	16	21	
Mean SHAP	Tree	**1**	**2**	13	**4**	9	**3**	18	6	**5**	10	12	11	8	7	15	16	
	Hierarch.	**1**	**3**	13	27	7	**2**	15	6	**4**	9	10	12	**5**	8	11	25	
	Interval	**1**	**2**	13	**5**	8	**3**	17	10	**4**	9	11	12	6	7	15	16	
	K-means	**1**	**4**	11	27	9	**3**	14	**5**	**2**	7	10	12	6	8	15	25	
Main SHAP	Tree	**1**	**3**	12	**5**	7	**2**	16	8	**4**	11	9	10	6	14	17	19	
	Hierarch.	**1**	**3**	12	26	6	**2**	14	7	**4**	9	8	10	**5**	13	15	25	
	Interval	**1**	**3**	12	**5**	7	**2**	17	9	**4**	11	8	10	6	14	16	18	
	K-means	**1**	**3**	12	25	7	**2**	14	6	**4**	9	8	10	**5**	13	16	26	
Slope SHAPs	Tree	**1**	**2**	7	**4**	9	**3**	6	**5**	13	10	8	14	12	11	17	18	
	Hierarch.	**1**	**5**	**3**	15	6	**2**	**4**	7	11	16	8	12	9	10	13	19	
	Interval	**1**	**3**	**2**	**5**	9	**4**	8	6	14	7	11	13	10	12	18	17	
	K-means	**1**	**3**	**4**	13	**5**	**2**	6	7	9	16	11	14	10	8	15	21	

Weight	9	**3**	18	13	**1**	8	22	11	**4**	6	**2**	**5**	7	12	17	14	*GFI*
Gain	**2**	**3**	**1**	14	6	9	**4**	8	15	**5**	17	16	10	7	23	21	
Cover	**2**	8	**1**	**3**	14	6	**4**	**5**	10	9	19	16	7	11	18	21	
Total Gain	**1**	**2**	**4**	14	**3**	6	16	10	8	**5**	9	12	7	11	22	17	
Total Cover	**1**	**3**	**5**	**2**	7	6	15	8	**4**	10	11	12	9	13	20	17	
Abs. Mean SHAP	**1**	**4**	**2**	6	**5**	**3**	8	9	12	13	10	14	11	7	18	19	

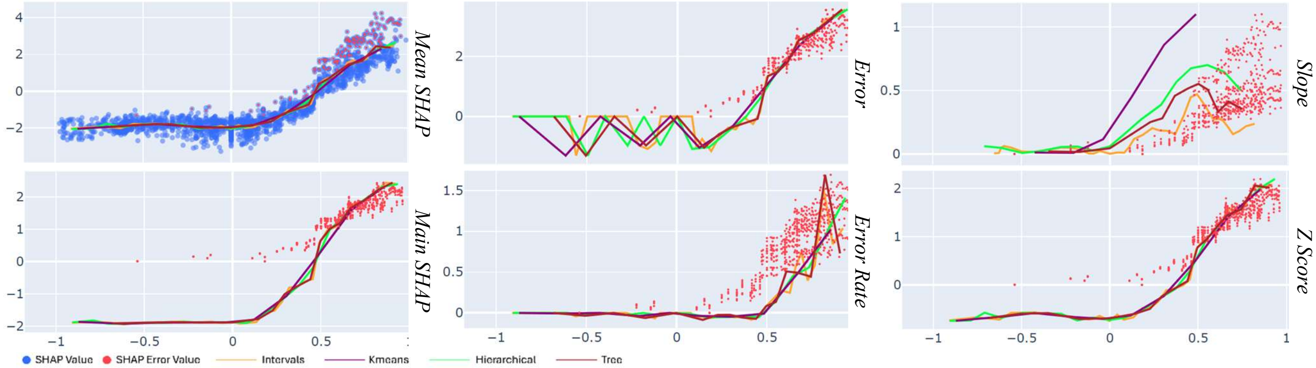


Figure 3. “Orientation_Index” The feature significantly influences error prediction within the range of [0.5, 1]. Main SHAP values indicates importance, Error SHAP indicates a strong association with error causes, and Error Rate SHAP underscores the feature’s relevance in regions with few data points but substantial error instances. Finaly, the Slope SHAP metric reveals a steep increase over the feature range [0, 0.5], where SHAP values exhibit a sharp incline.

A. Features with Global and Regional Importance

The following discussion presents the results about features which are important based on both GFI and RFI.

The GFI ranks Orientation_Index, as shown in Table 4, as the first and second most important feature, based on Total Gain, Total Cover, Gain, and Cover. This indicates that Orientation_Index on average impacts many data points in its splits which lead to substantial improvement in performance. In addition, Mean Abs. SHAP also places it on the first rank, highlighting the feature’s overall contribution to the prediction.

The RFI measures allow a more nuanced picture. Mean SHAP ranks Orientation_Index as first and second most important, showing its positive influence in the [0.5, 1] range, as shown in Figure 3. Main SHAP metrics provides an indication of how much the feature itself (rather than interactions with other features) can be considered as error explanation. The increased Main SHAP values in the interval [0.5,1] suggest that the feature is an interesting candidate for explaining errors in this region. Error SHAP quantifies the average impact of error instances. Error SHAP ranks it high across all partitioning methods, indicating a strong relation to error causes in the [0.5, 0.9] region. Error Rate SHAP ranks the feature as important on all partitioning methods. This aggregation considers the mean SHAP value and the error rate, focusing on regions with small numbers of data points with a significant amount of error instances. Error Rate SHAP shows increased values in the region of [0.5, 1], indicating separable errors for which the feature serves as a suitable explanation. The Slope SHAP metric exhibits a steep increase over the feature range of [0,0.5], where the feature shows a sharp increase of SHAP values. This indicates that the feature has a significant impact on the prediction of errors above the threshold of 0.5. Z-Score SHAP ranks the feature highly, indicating above-average impact. Overall, the RFI indicates that Orientation_Index is an interesting candidate to consider as explanation of error cause for the region above a feature value of 0.5.

GFI ranks Length_Of_Conveyer (Figure 4) as second to fourth, showing its frequent use and high impact. RFI also places it in the top five, with Mean SHAP showing increased values in [1690, 1700] and Main SHAP identifying it as the main driver. Error SHAP highlights strong relations to errors

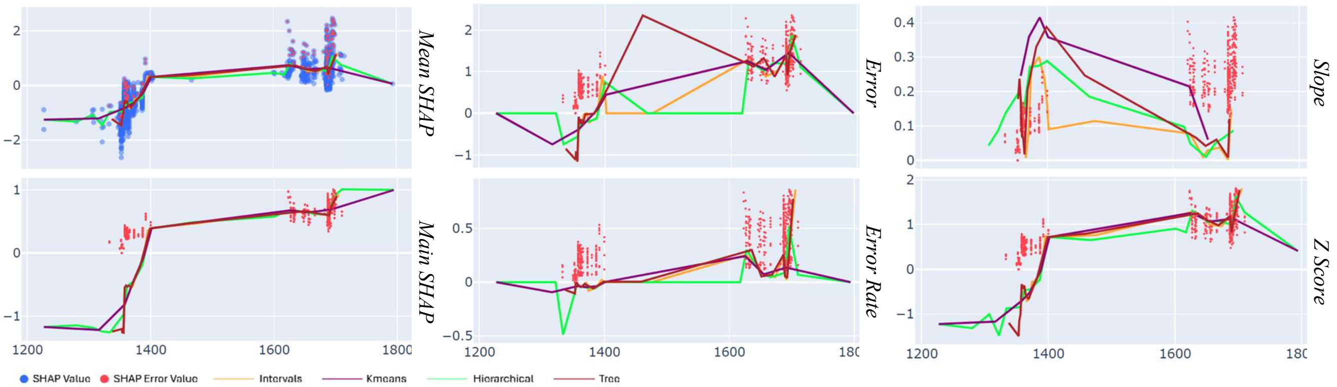


Figure 4. “Length_of_Conveyer”: Mean SHAP reveals increased SHAP values within the range of [1690, 1700] and Main SHAP values suggest that the feature itself is the primary driver in this region. Error SHAP emphasize a strong association with errors in the region [1600, 1700]. Additionally, Slope SHAP identifies a threshold, and Z-Score SHAP indicates an impact higher than on average above a feature value of 1400.

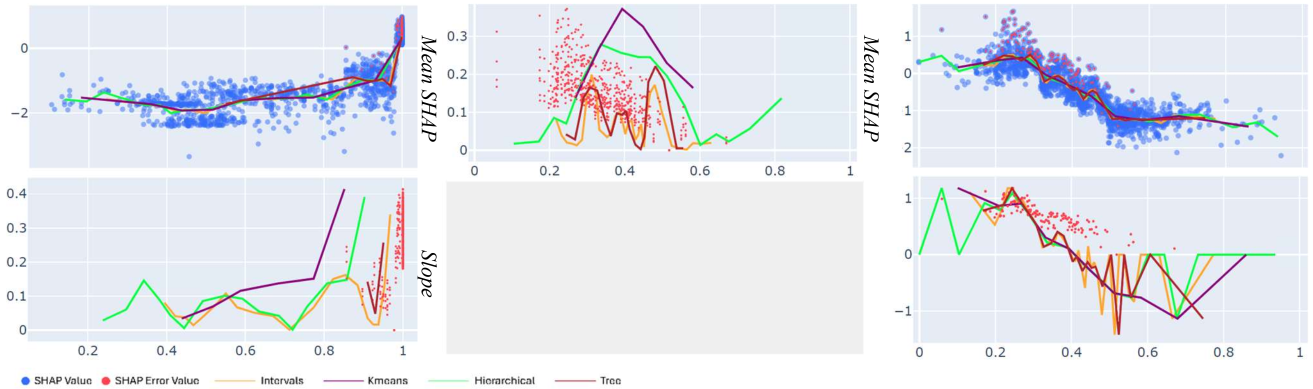


Figure 5. “Edges_Y_Index and Empty_Index”: Edges_Y_Index (left) exhibits high importance in the Slope aggregation. The slope of this feature sharply increases in the region [0.9, 1]. Quality engineers can leverage this threshold to adjust alarms. Empty_Index (right) shows increased relations to errors in the interval [0.2, 0.3] indicated by Error SHAP.

in [1400, 1700], and Error Rate SHAP points to a critical error rate at 1700. Z-Score SHAP confirms its above-average impact above 1400, suggesting its critical role in product quality at high values.

The GFI measures rank Edges_Y_Index as the most important feature, highlighting its broad impact. Additionally, Mean Abs. SHAP also ranks the feature as the most important, indicating a substantial impact on the model’s overall performance. The Slope SHAP aggregation ranks Edges_Y_Index high. Slope SHAP measures the change in mean SHAP values over the feature value range. Specifically, as shown in Figure 5, the slope is high in the region [0.9, 1], where Edges_Y_Index has a sharp increase of SHAP values. This presents a threshold where quality engineers could adjust production alarms to.

The GFI measure ranks Empty_Index as the first and the third most important feature, based on Weight and Total Gain. This shows that the feature is often used as split criteria, although it leads to improvements in prediction only in a few instances. The Mean Abs. SHAP ranks the feature fifth, indicating that it decently contributes to the overall model predictions. The RFI ranks Empty_Index (Figure 5) on Error SHAP and Slope SHAP as the fifth most important feature, according to the K-mean and the Tree partitioning method. The feature shows an increased contribution of errors in the interval [0.2,0.3] with decreasing effects up to 0.4 where the Slope SHAP also shows a threshold. Leveraging this threshold for production alarms could be valuable.

The GFI measures rank Outside_X_Index as the second and third most important feature, based on Total Cover and Cover. These indicate that the feature influences a large number of data points both in total and on average when used in the splits. The RFI, as shown in Figure 6, reveals that for error causes analysis, the interval [0, 0.01] is particularly interesting. The Mean SHAP in this region shows increased values, supported by the Error Rate metric, which shows an increased error rate in this region. Furthermore, the Main SHAP values are also increased, suggesting that Outside_X_Index itself is the primary driver of the effects. The Slope aggregation displaying the change of SHAP values shows a steep decrease, potentially indicating a threshold leading to actionable insights. Additionally, the Z-Score SHAP indicates above average performance of feature values <0.007. This shows that Outside_X_Index is important for explaining errors in production for small feature values. The GFI ranks Minimum_of_Luminosity as the fourth most

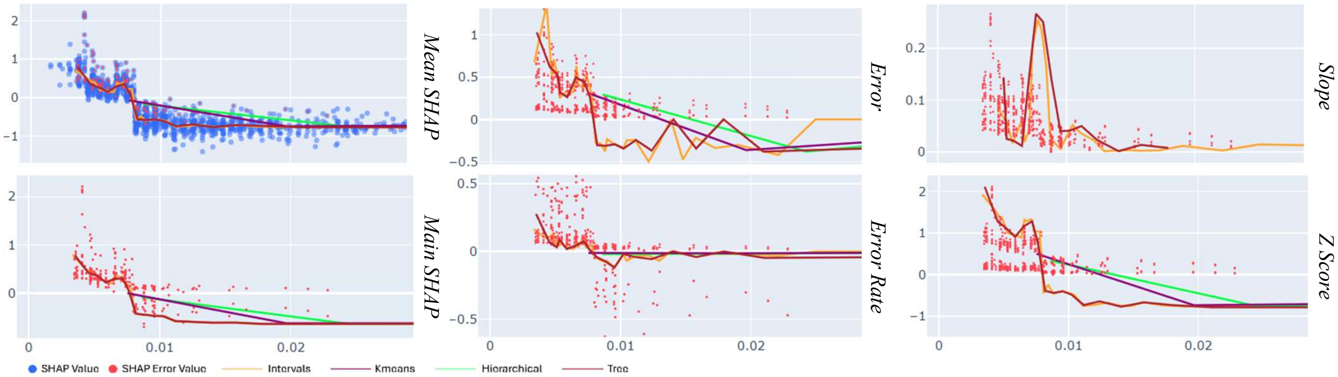


Figure 6. “Outside_X_Index”: The interval [0, 0.01] is particularly interesting. Within this region, Mean SHAP values increase, supported by the Error Rate metric, which indicates an increased error rate. Moreover, Main SHAP values are higher, suggesting that Outside_X_Index itself drives the resulting effects. Slope SHAP displays a steep decrease, potentially indicating a threshold for actionable insights. Additionally, the Z-Score SHAP indicates above average performance of feature values < 0.0075.

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

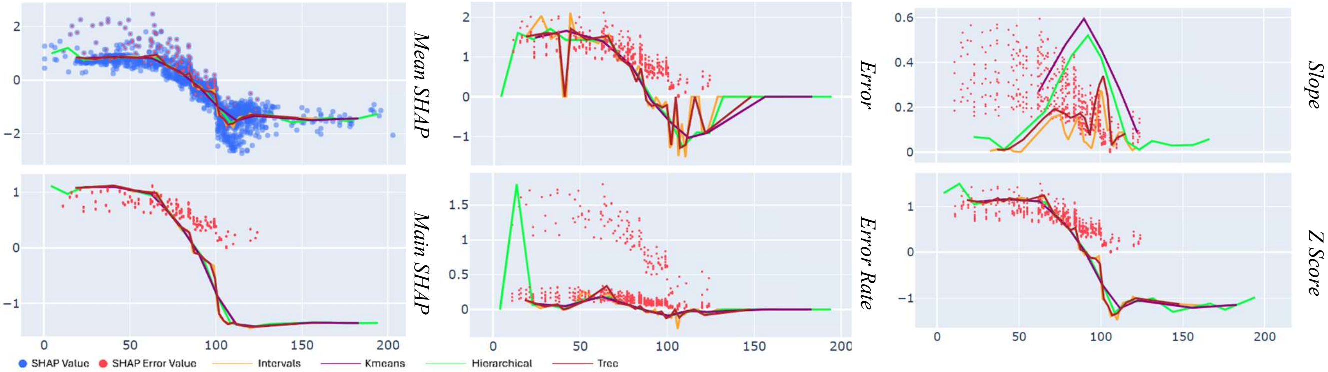


Figure 7. "Minimum_of_Luminosity": An important feature for explaining error causes, especially in the region of [0, 75]. In this region, the feature has a positive impact on predicting errors, and not because of interactions. The feature shows a high error rate and a higher impact than on average in this region. The threshold of 75 could be a potential threshold for detecting errors.

important feature, based on Mean Abs. SHAP. This indicates a high impact over the entire feature value range for both error and non-error cases. The RFI ranks Minimum_of_Luminosity, as shown in Figure 7, second and third on Mean, second on Main, second and third for Error, second to fifth for Error Rate, and third and fourth rank for the Slope SHAP metrics. Specifically, Mean SHAP shows a positive influence on predicting errors in the region of [0, 75], highlighting its critical role for error analysis. In the same [0,75] region, the Main SHAP metric shows increased values, suggesting that the feature itself drives errors rather than interactions. The Error SHAP metric shows increased value for Minimum_of_Luminosity in the region of [0, 75], implying that the feature is more likely to cause errors in this region.

For feature values smaller than 25, the Error Rate SHAP shows increasing error rates, indicating a sparse region with high error ratio. The increased Z-score SHAP values for the region [0,75] emphasizes its impact in this region beyond the feature average. The Slope SHAP shows a sharp change of SHAP values which suggests a potential threshold for a feature value of 75. Overall, the RFI underscores that Minimum_of_Luminosity is a critical factor for the quality of products. It implies that the feature is more likely to serve as explanation in the region [0,75] with potential threshold of around 75 for identifying errors.

The GFI measures ranks Log_X_Index as one of the top five most important features, based on Gain and Cover. This indicates that the feature produces splits that affect a great number of data points and have a great impact on the prediction. For the RFI, as shown in Figure 8, Slope SHAP ranks the feature on fourth place on the hierarchical partitioning method and shows a threshold at a feature value of 1.25. The minor importance assigned from others RFIs points to the negative SHAP value above the feature value of 1.25, suggesting that feature is indicative of non-error instance above that threshold.

The GFI ranks Steel_Plate_Thickness (Figure 8) as the fifth most important feature, based on Cover. This indicates that the feature splits affect a large amount of data. However, the RFI reveals the relations to error causes. The error rate aggregation ranks the feature high for the hierarchical and k-means partitioning, showing that the error rate of Steel_Plate_Thickness is higher in the region of [100, 200]. The Z-Score SHAP ranks this feature on fourth, indicating a greater-than-on average impact in the region. The Slope

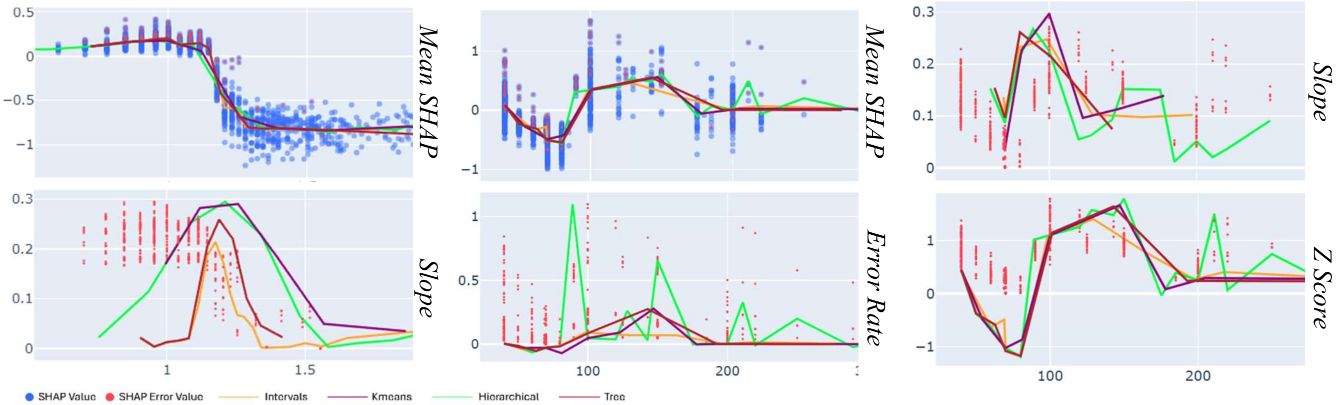


Figure 8. "Log_X_Index and Steel_Plate_Thickness": Log_X_Index (left) is of minor interestingness for error explanation, except the threshold of around 1.25. Steel_Plate_Thickness (mid and right) exhibits and increased error rate and greater than average impact in the region of [100, 200], as indicated by Error Rate and Z-Score. The Slope SHAP also shows a threshold of around 100.

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

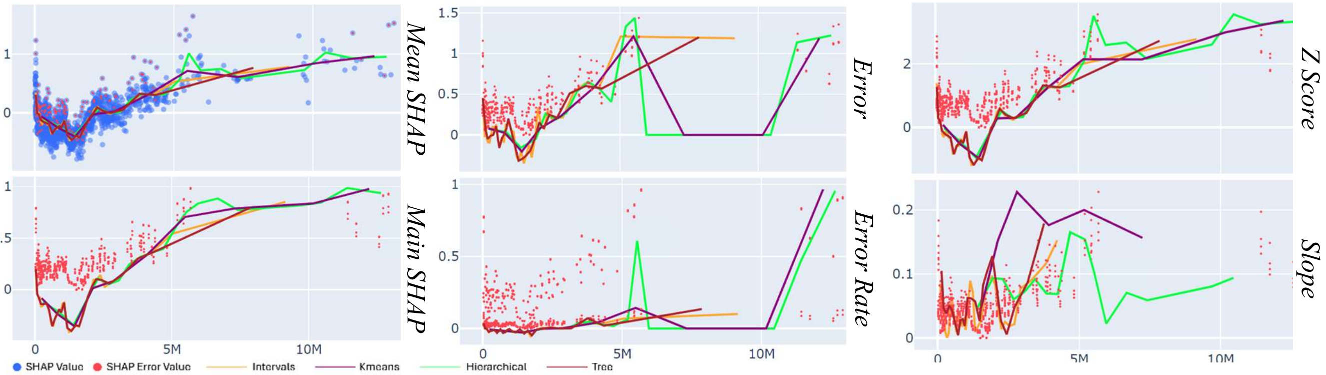


Figure 9. "Y_Minimum": The feature has an increasing impact on predicting errors in the region of [5M, 15M], and not because of interactions. It shows a high error rate and a higher than average impact in the region of >10M. Slope SHAP suggests some thresholds for <5M. Y_Minimum is a key factor for error analysis, especially in the regions of >5M.

metric indicates a threshold at a feature value of about 100. Consequently, Steel_Plate_Thickness is likely a critical factor in the range [100, 200], with increased error frequency and sufficient SHAP effects.

Feature Y_Minimum, as shown in Figure 9, is ranked as the fourth most important feature, based on Weight and Total Cover. This indicates a frequent use and many influenced data points, at least in some splits.

The Mean SHAP ranks Y_Minimum on second to fifth place, driven by positive effects in the region [5M, 15M]. This implies a critical influence on the product in this region. Main SHAP ranks the feature on fourth, showing increasing influence of the feature from [2M, 15M]. This indicates that the influence is caused by the feature itself. The Error SHAP metric ranks the feature in fourth and sixth place, showing increasing influence in the interval [2M, 7.5M] and above 10M. Error SHAP provides an indication of contribution to errors in products in these regions. The error rate SHAP metric ranks Y_Minimum high for the k-means and hierarchical partitioning. These partitioning methods focuses on separated regions, which also reveal patterns in sparse regions. The error rate is higher in the region of >10M, implying clearly separable product errors. Moreover, the Z-Score SHAP shows increasing values, indicating an impact greater than average. Therefore, Y_Minimum is interesting for error analysis and likely to explain the errors in the region of a feature value <5M.

The GFI ranks X_Minimum, as shown in Figure 10, as one of the five most important features, based on Gain and Total Gain. This suggests that using the feature as splitting criteria improves the predictive performance both in total and on average. The RFI ranks the feature as important on the Error Rate and the Z-Score metrics. Specifically, the feature shows an increased error rate at a feature value of about 1500, indicating separable errors with high frequency and a strong association with errors in that region. The Z-Score SHAP ranks the feature third for the K-means partition method, indicating an effect greater than average in this region. Consequently, X_Minimum is a promising candidate for explaining error on high feature values.

B. Features with Global Importance

In the following, we focus on the features that are ranked as important only GFI measures, but not by the RFI. The GFI ranks Edges_Index (Figure 11) as the second most important feature, based on Weight. This indicates that Edges_Index is frequently used in the splits of the model. However, the regional feature importance measures reveal that the feature is not particularly useful for explaining error causes. The SHAP values over the entire feature range have small effects. Moreover, the feature is often used as indicated by Weight, however, without effect on the predictive performance. The GFI rank Luminosity_Index (Figure 11) as the fifth important feature, based on Weight. This indicates that the feature is frequently used in the splits of the model. Luminosity_Index is not ranked as important from the RFI. The feature is placed on rank twelfth for Mean SHAP and tenth on the Main SHAP metrics, indicating some influence of the feature to the model

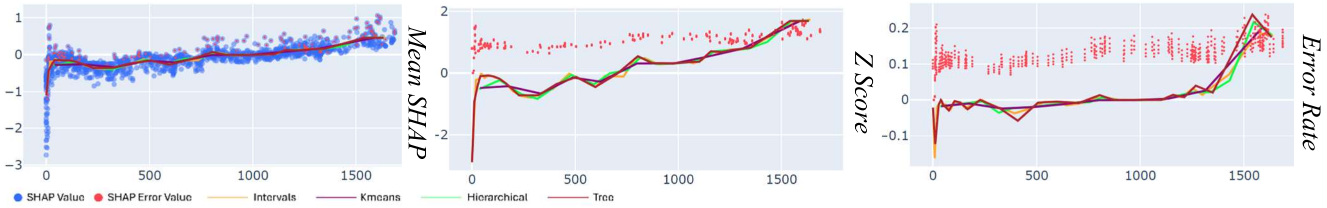


Figure 10. "X_Minimum, Luminosity_Index and Edges_Index": X_Minimum (left and mid) shows a high importance in the region around 1500, where the Error Rate and the Z-Score SHAP indicates a strong association with errors and a greater than average impact.

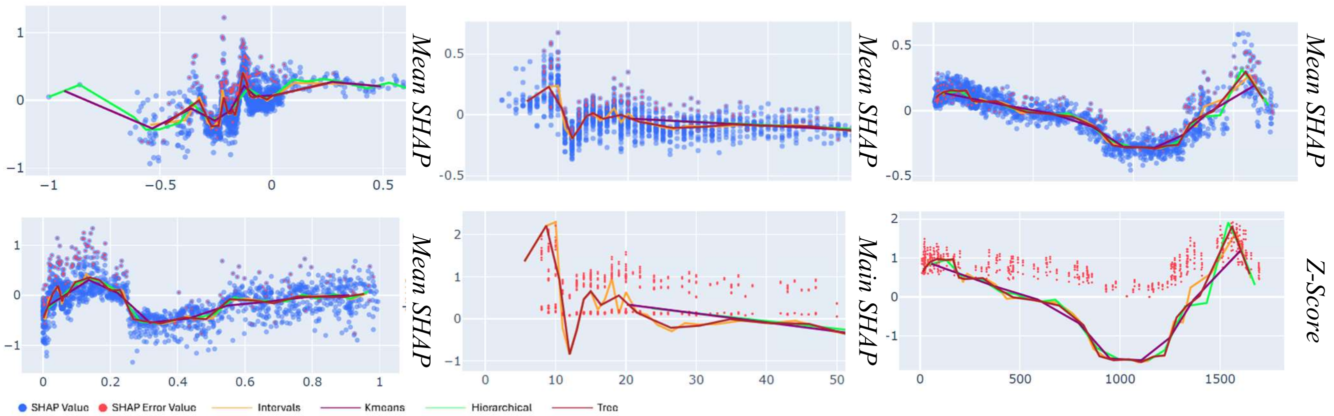


Figure 11. “X_Perimeter, Edges_Index, Luminosity_Index and X_Maximum”: Edges_Index (top left) and Luminosity_Index (bottom left) are both important, based on Weight. The SHAP values of both features are small and do not show any clear patterns. Therefore, both feature are not very interesting. X_Perimeter (mid) shows increased main effects in the region [8, 10], indicating that the feature itself primarily causes the effect. X_Maximum (right) shows increased Z-Score SHAP values for feature values greater than 1500, indicating a higher than average impact of the feature in this region.

performance. However, due to its weak effects, this feature is not interesting for error analysis.

C. *Features with Regional Importance Only*

In the following, we focus on the features that are ranked in the top five by the RFI measures, but not by the GFI measures. These features are interesting because they reveal the limitations of the GFI measures in capturing the local patterns and behaviors of the features that are relevant for error analysis.

The GFI do not rank X_Perimeter, as shown in Figure 11, in the top five. The Z-Score SHAP on the tree and interval partitioning methods show increased effects in the interval [8,10]. However, Main SHAP ranks the feature with minor importance which shows that these effects are resulting from interactions with other features. The GFI do not rank X_Maximum (Figure 11) in the top five. The Z-Score SHAP on the tree and hierarchical partitioning methods show increased effects for feature values greater than 1500. Error Rate SHAP ranks the feature eighth and ninth, indicating some separable errors. However, Main SHAP ranks the feature with minor importance which shows that these effects also result from interactions.

The GFI do not rank Maximum_of_Luminosity, as shown in Figure 11, in the top five. However, the RFI rank the feature on the fifth most important. Error SHAP for the tree and the interval partition method indicate increased SHAP values of error instances on the interval [125, 200]. In the same region, the Main SHAP metrics also show that the feature is likely the primary driver of the effect. The increased Z-Score SHAP metric indicates an impact greater-than-average in the region. Therefore, Maximum_of_Luminosity is an interesting feature for error analysis.

The GFI does not rank Square_Index (Figure 12) in the top five. However, Error Rate SHAP ranks the feature fourth for both the tree and interval partitioning methods, indicating

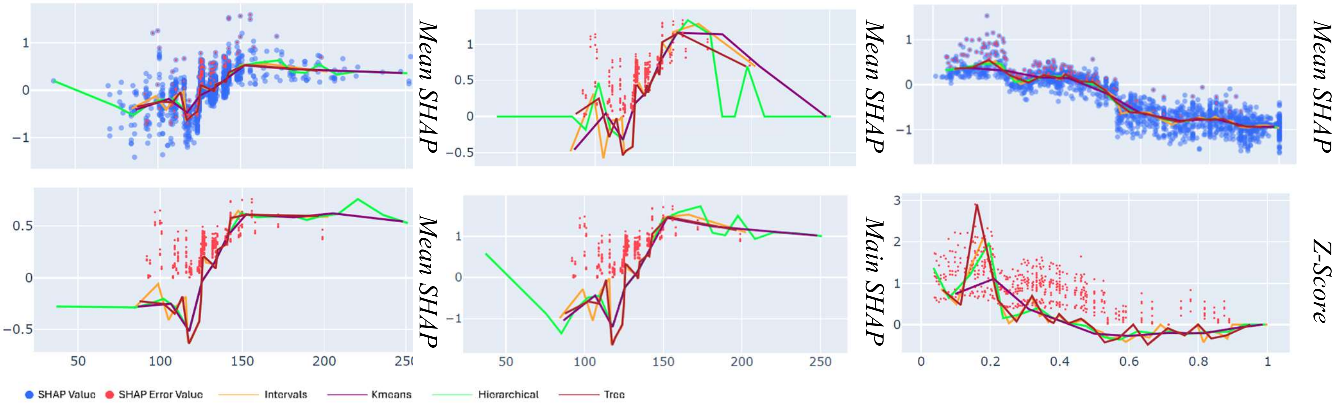


Figure 12. “Maximum_of_Luminosity and Square_Index”: Maximum_of_Luminosity (left and mid) is a critical factor for the error analysis within the region [125, 200]. Error SHAP exhibits increased values, Main SHAP indicates that the feature itself influences the effect, with an impact greater than on average, as indicated by Z-Score SHAP. Square_Index (right) shows increased error SHAP in the region of [0.1, 0.2], which indicates a separated errors and identifies the feature as an interesting candidate for explaining errors in this region.

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

an increased error rate in the region of [0.1, 0.2]. This implies that the feature has separable regions with higher error frequency. Consequently, the feature is an interesting candidate for explaining these errors.

In summary, we evaluated the RFI on the dataset from the steel manufacturing domain using an XGBoost model. The RFI measures, applied across 20 partitions for each feature, provided insights that GFI measures could miss, particularly in identifying specific feature value intervals crucial for error prediction. This nuanced understanding of regional effects is vital for actionable insights in error analysis. The analysis revealed common assignments of importance across features comparing GFI and RFI. This shows that RFI captures the notion of importance as embedded by the GFI, which is global, or model performance-based. Moreover, the RFI revealed interesting features, and the provided interesting insights about patterns that the GFI missed. Understanding these regional nuances can lead to actionable insights for error analysis in manufacturing.

VI. RELATED WORK

Root cause analysis in the production environment has been well studied [19] and several methods for model interpretability through XAI have been reported [20]. However, we argue that the proposed metrics are more related to feature importance measures. The metrics may be used in root cause analysis to incorporate expert knowledge. Applied XAI in the manufacturing domain is used to extract explanations from a machine learning model to, e.g., enhance trust in the model, used for model optimization or to assist domain experts. In [21] saliency maps and class activation maps are extracted from a deep learning model. In [4] the authors use an isolation forest as model to determine normal production line behavior and feature importance to explain the model. Mehdiyev and Fettke apply local and global explanations to examine the impact of different views on the generated insights [14]. However, neither work addresses the problem of which feature provides the most promising insights given the possible tremendous feature space and the corresponding effort required to examine all explanations. To the best of our knowledge, we are the first to provide SHAP-based importance measures tailored to the task for quality management.

Lundberg et al. introduced the idea of SHAP-based feature importance [13]. To determine a feature's overall effect the absolute SHAP value across all considered instances is averaged and thus a global importance measure. In contrast, our proposed measures just consider instances that possibly encompass interesting properties for quality management. Other global importance measures used in the domain have a broad history. A detailed description of the following global importance measures is laid out by Molnar [22]. In [23] Permutation Feature Importance is introduced. A global measure where the features are perturbed and the resulting performance loss of the model is taken as a measure of the feature's importance. Mehdiyev and Fettke [18] used Individual Conditional Expectation (ICE) [24] as the global importance. Another method possibly used are Partial Dependence Plots (PDP) [25]. However, neither ICE nor PDP accumulates a single importance score. Both are used as visualizations of global model behavior.

Overall, one of the most influential global importance measures is the Gini index [26]. According to Lundberg [27], the Gini index is equivalent to the in XGBoost [17] implemented importance measure Gain, which *uses the average training loss reduction gained when using a feature for splitting*. Lundberg [27] also describes Weight as *the number of times a feature is used to split the data across all trees* and Cover as *the number of times a feature is used to split the data across all trees weighted by the number of training data points that go through those splits.* Both the total importance scores used for comparison are described in the XGBoost documentation [28] for Total Gain as the *total gain across all splits the feature is used in* and Total Cover as *the total coverage across all splits the feature is used in.* For local feature importance also LIME [12] could be considered. However, to compute explanations LIME uses sampling which is not restricted to solely interesting areas.

VII. CONCLUSION

In this paper, we introduced RFI measures that aim at identifying interesting features for quality management in manufacturing. We discussed the underlying notion of interest and provided corresponding formal definitions. Conceptually, RFIs are between established global and local feature importance measures and highlight regional effects which are helpful in finding production error causes. We illustrate the usefulness of the new measures through experiments using synthetic and real-world data.

Our experiments show that the proposed measures provided detailed insights on features – based on our experience [5] – are interesting; moreover, are partly missed by established methods. Therefore, we conclude that with the help of the proposed importance measures, quality managers get hints about interesting relations that are reflected in the prediction model to drive deeper analysis. Thus, quality managers benefit from adding the proposed importance measures to the pool of XAI methods and we thereby improve XAI for error prediction in manufacturing.

Subject to future work are questions about the integration of RFI in the machine learning pipeline. We assume that the measures are applied at the end of the pipeline, potentially after feature engineering and model optimization. However, the proposed measures may drive the analysis of features earlier in the pipeline as well. Additionally, future work may expand the range of partitioning and aggregation methods to enhance the detection of complex error patterns. Investigating the measures in larger datasets and diverse manufacturing settings could further validate and refine their applicability.

REFERENCES

[1] V. Göttisheim, H. Ziekow, U. Schreier and A. Gerling, "Shapley Values based Regional Feature Importance Measures Driving Error Analysis in Manufacturing", *DATA ANALYTICS 2022, The Eleventh International Conference on Data Analytics*, November, 13 - 17, 2022, Valencia, Spain, pp. 19-26, 2022.

[2] C. Seiffer, A. Gerling, U. Schreier and H. Ziekow, "A Reference Process and Domain Model for Machine Learning Based Production Fault Analysis", *Enterprise Information Systems: 22nd International Conference, ICEIS 2020,* Springer International Publishing, pp. 140–157, 2021.

[3] A. Gerling et al., "Comparison of algorithms for error prediction in manufacturing with automl and a cost-based metric", *Journal of Intelligent Manufacturing*, 33.2022(2), pp. 555–573, 2022.

[4] H. Ziekow, U. Schreier, A. Gerling and A. Saleh, "Interpretable Machine Learning for Quality Engineering in Manufacturing - Importance Measures that Reveal Insights on Errors", *The Upper-Rhine Artificial Intelligence Symposium, UR-AI 2021, Artificial Intelligence - Application in Life Sciences and Beyond*, Germany, Kaiserslautern: Hochschule Kaiserslautern, University of Applied Sciences, pp. 96–105, October 2021

[5] M. Carletti, C. Masiero, A. Beghi and G.A. Susto, "Explainable Machine Learning in Industry 4.0: Evaluating Feature Importance in Anomaly Detection to Enable Root Cause Analysis", *IEEE International Conference 2019*, pp. 21–26, 2019.

[6] H. Ziekow et al., "Proactive Error Prevention in Manufacturing Based on an Adaptable Machine Learning Environment", *Artificial Intelligence: From Research to Application: The Upper-Rhine Artificial Intelligence Symposium UR-AI 2019*, Offenburg, Germany, Karlsruhe: Hochschule Karlsruhe - Technik und Wirtschaft, pp. 113–117, March 2019.

[7] R. S. Peres, J. Barata, P. Leitao and G. Garcia, "Multistage Quality Control Using Machine Learning in the Automotive Industry", *IEEE Access*, vol. 7, pp. 79908–79916, 2019.

[8] Y. Wilhelm, U. Schreier, P. Reimann, B. Mitschang and H. Ziekow, "Data Science Approaches to Quality Control in Manufacturing: A Review of Problems, Challenges and Architecture", *Symposium and Summer School on Service-Oriented Computing*, Springer, pp. 45–65, 2020.

[9] A. Gerling et al., "Results from using an Automl Tool for Error Analysis in Manufacturing", *Proceedings of the 24th International Conference on Enterprise Information Systems - Volume* 1, pp. 100–111, 2022.

[10] C . Seiffer, A. Gerling, U. Schreier and H. Ziekow, "A Reference Process and Domain Model for Machine Learning Based Production Fault Analysis", *Enterprise Information Systems: 22nd International Conference, ICEIS 2020*, Springer International Publishing, pp. 140–157, 2021.

[11] S. M. Lundberg and S. I. Lee, "A unified approach to interpreting model predictions", *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*, NY, USA, pp. 4768–4777, 2017.

[12] M. T. Ribeiro, S. Singh and C. Guestrin, "Why Should I Trust You?": Explaining the Predictions of Any Classifier", *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, pp. 1135–1144, 2016.

[13] S. M. Lundberg et al., "From local explanations to global understanding with explainable AI for trees", *Nature Machine Intelligence, 2(1)*, pp. 56–67, 2020.

[14] N. Mehdiyev and P. Fettke, "Local Post-Hoc Explanations for predictive Process Monitoring in manufacturing", *29th European Conference on Information Systems - Human Values Crisis in a Digitizing World, ECIS 2021,* Marrakech, Morocco, 2020.

[15] M. Saarela and S. Jauhiaine, "Comparison of feature importance measures as explanations for classification models", *SN Appl. Sci. 3*, p. 272, 2021.

[16] G. Casalicchio, C. Molnar, and B. Bischl. "Visualizing the Feature Importance for Black Box Models". In: *Berlingerio, M., Bonchi, F., Gärtner, T., Hurley, N., Ifrim, G. (eds) Machine Learning and Knowledge Discovery in Databases. ECML PKDD 2018. Lecture Notes in Computer Science*, vol 11051. Springer, Cham, 2019.

[17] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System", *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA: ACM, pp. 785–794, 2016.

[18] M. Buscema, S. Terzi and W. Tastle, "Steel Plates Faults " *UCI Machine Learning Repository*, Irvine, CA: University of California, School of Information and Computer Science, 2010. [Online]. Available from: https://doi.org/10.24432/C5J88N.

[19] E. Oliveira, V. L. Miguéis and J. L. Borges, "Automatic root cause analysis in manufacturing: an overview & conceptualization", *Journal of Intelligent Manufacturing*, 2022.

[20] G. Sofianidis, J. M. Rožanec, D. Mladenić and D. Kyriazis, "A Review of Explainable Artificial Intelligence in Manufacturing", *Trusted Artificial Intelligence in Manufacturing: A Review of the Emerging Wave of Ethical and Human Centric AI Technologies for Smart Production*, pp. 93–113, 2021.

[21] C. V. Goldman, M. Baltaxe, D. Chakraborty and J. Arinez, "Explaining Learning Models in Manufacturing Processes", *Procedia Computer Science, 180*, pp. 259–268, 2021.

[22] C. Molnar, "Interpretable Machine Learning: A Guide for Making Black Box Models Explainable", *2nd edn.*, 2022. [Online]. Available from: https://christophm.github.io/interpretable-ml-book, retrieved on 08/26/2022.

[23] L. Breiman, "Random Forests", *Machine Learning 45*, pp. 5–32, 2001.

[24] A. Goldstein, A. Kapelner, J. Bleich and E. Pitkin, "Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation", *Journal of Computational and Graphical Statistics,* 24, pp. 44–65, 2015.

[25] J. H. Friedman, "Greedy function approximation: A gradient boosting machine", *The Annals of Statistics, 29 (5),* pp. 1189–1232, October 2001.

[26] T. Hastie, R. Tibshirani and J. Friedman, "Random forests", *The Elements of Statistical Learning*, Springer, pp. 587–604, 2009.

[27] S. M. Lundberg, "Interpretable Machine Learning with XGBoost", April, 2018. [Online] Available from: https://towardsdatascience.com/interpretable-machine-learning-with-xgboost-9ec80d148d27, retrieved on 10/06/2024.

[28] XGBoost Documentation, "Python API", *Reference. xgboost developers*. [Online]. Available from: https://xgboost.readthedocs.io/en/latest/python/python_api.html, retrieved on 08/26/2022

Transformation of Entity Relationship Models to Relational Models - A Practical Approach

Gregor Grambow
Aalen University
Aalen, Germany
Email: gregor.grambow@hs-aalen.de

Simon Ruttmann
eXXcellent solutions GmbH
Ulm, Germany
Email: simon.ruttmann@exxcellent.de

***Abstract*—Both Entity Relationship (ER) modeling and the relational model have come a long way and are both prevalent. The former for creating conceptual database schemata and the latter for creating technical database schemata. Unfortunately, both of these models are not directly compatible. Over the decades, various theoretic approaches for transforming ER models to relational models have been proposed. These efforts, however, did not yield practical approaches and thus editors capable of such transformations. Modern editors either have no transformation facilities, or do not use the ER model proposed by Peter Chen but rather provide somehow enhanced database diagrams. Therefore, users have to manually transform ER models to relational models, which is cumbersome and error-prone. To tackle this problem, we propose an automatic transformation from ER models to relational ones that focuses on practical applicability and operational semantics. In particular, the approach enables transformations of the original ER model by Chen as well as prevalent extensions. We have proven the applicability of this approach in two ways: First, we have created a graphical editor capable of flexibly modeling ER diagrams and automatically transforming them to relational models. Second, we have conducted a practical study with our approach and the created editor. Our findings show, that the transformation approach works correctly and the implemented editor makes it accessible to users.**

Index Terms—Entity Relationship; ER Model; Relational Model; Database; Editor

I. INTRODUCTION

This article is an extension to the conference paper ”A Practical Automated Transformation of Entity Relationship Models to Relational Models” published in 2023 [1]. Entity Relationship (ER) modeling is a prevalent option for semantic data modeling primarily applied to database schemata. This way of modeling has been used since over 40 years. It was introduced in 1976 by Peter Chen [2] and has been the focus of active research for decades. There has been a myriad of extensions to the model, like the ECR model [3], the ECR+ model [4], HERM [5], or the EER model [6]. Many of these extensions offer valuable additions to the basic ER models. Some features have been adopted but many have also been discarded. As a result of this, basic ER modeling became prevalent but with a high number of different flavors. The usage of generalization concepts, cardinality constraints or the application of the n-ary relationships as proposed by Peter Chen differs in many applications.

However, to be usable in a relational database, ER models have to be converted to relational models. Manually executed, this process can be tedious and error-prone. Thus, various approaches for standardized transformations have been proposed, like [3] [7] - [13]. Most of these approaches have two major downsides: First, they often impose certain constraints on the ER models and second, they remain rather theoretic. In many cases, after presenting their approach, the authors recommended them to be used by practitioners or in automated tools. Despite having a clear formalism, a practical implementation was never achieved, often due to the lack of operational semantics. On the other hand, there is a high number of editor tools for ER models, contained in drawing tools [14] - [17], in client software of databases [18] [19], low-code plattforms [20] [21], or in modeling tools like Enterprise Architect [22].

These editors have two main issues: Each of them uses a different subset of ER concepts, some are even closer to a relational or even UML class diagram editor. In addition, the transformation aspect has been ignored almost completely. Thus, semantic data modeling for databases usually involves first drawing an ER diagram and then manually transforming it to database tables, which can be a source of numerous issues.

Contrary to the above mentioned tools the proposed approach stands out for its depth and flexibility when it comes to dealing with ER diagrams. It can capture complex relationships between entities, relations and a variety of attribute types. This enables a more accurate representation of the underlying data structure, resulting in better maintainable and more efficient databases.

To counteract the mentioned transformation issues, we propose an approach for automatically transforming ER models to relational models. As opposed to prior approaches we do not focus on mathematical definitions or calculus but rather on practical applicability and operational semantics. Thereby, our approach can be easily applied to editors in different programming languages. To prove this applicability we have implemented a graphical ER editor that is capable of this automatic transformation.

The most significant aspect of the proposed algorithms and presented tool is thus removing the gap between real ER models and relational models. The proposed algorithms enable modelers to perform work on both the conceptual and technical level. As business requirements can be clearly articulated on a purely conceptual level the proposed approach provides a way for better understanding and representing the data’s

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

semantic structure. This contributes to more precise and better comprehensible relational models.

There are two main extension points of this article to the conference paper "A Practical Automated Transformation of Entity Relationship Models to Relational Models" published in 2023 [1]. Firstly, within the practical implementation section, the validation mechanisms performed during the er modeling process, which are developed and integrated into the modeling tool are also described extensively. The second point of extension is the evaluation of the applicability of the developed modeling tool. The evaluation focuses on comparing the ER modeling process experience utilizing the presented tool against other prevalent tools.

The rest of this paper is organized as follows: Section II discusses related approaches, while Section III defines the concrete style of ER diagram that is assumed as basis for the transformation. Section IV provides an extensive description of the transformation approach. After that, Section VI shows a practical evaluation of the proposed approach followed by Section VII providing a conclusion as well as future directions.

II. RELATED WORK

In this section, we cover two types of related work: Scientific approaches presented for transforming ER models into relational models and practical ER editor tools.

In the scientific community, the case of transforming ER models has been extensively discussed over the decades. Most proposed approaches date back to the 1980s and 1990s. As mentioned, one big issue concerning general applicability is the high number of different ER variants. The basic variant proposed by Peter Chen [2] includes n-ary relationships and weak entities but no generalization concepts. Most of the extensions [3] - [6] to that model focused on adding structures for generalization. Due to that, many different transformation approaches take different variants of the ER model as basis. Most attention was paid to the original version of the ER model [7] - [10] and extended ER models with generalization structures [3] [11] - [13]. All of them have in common that they remain rather theoretic and do not consider operational issues [23]. Further, to the best of our knowledge, none of these approaches lead to the development of a prevalent ER editor tool.

The fact that most research regarding that topic was carried out decades ago lets us assume, that the transformation approaches have been adopted and are now prevalent in ER editors. Therefore, the second part of this section deals with contemporary ER editors. However, before investigating the transformation capabilities, another issue has to be dealt with: There is a high number of ER editors that do not use the ER model as proposed by Peter Chen. Many of them focus on simplified binary relationships. In addition, they do not cover the most prevalent extension to Chen's model: generalization. Such diagrams are often nearer to a relational database diagram than to a real ER diagram. One category featuring such diagrams is drawing tools like Lucid Chart [14], Draw.io [15], or Visual Paradigm [16]. Some of them even contain concepts like stored procedures or triggers and can thus not be considered ER editors. Furthermore, none of them provides a transformation approach. Another category providing such diagrams is database client software, e.g., from PostgreSQL [18] or MySQL [19]. In contrast to the database-specific editor tools there are also low-code platforms such as Mendix [21] and Outsystems [20] to mention. These tools also contain editors to make it possible to model relational models. The above listed tools provide visual modeling that can be directly used as database tables.

However, such diagrams are rather close to the relational approach and not to Chen's model. The number of editors covering the latter is rather limited. One with a good set of concepts is the Enterprise Architect [22], which covers most elements considered prevalent in ER models as of today: n-ary relationships, generalization, and multi-valued or composite attributes. However, there is no transformation approach in place. Only one editor features an ER model like proposed by Chen and also a transformation approach: ERD+ [17]. But that editor only features a rather limited set of modeling elements. It does not support n-ary relationships and the use of generalization is rather restrictive. In addition, the translation of weak types is only possible on the most trivial level. This also applies to the translation of multi-valued and composite attributes. The combination of attribute types is not supported at all. In summary, it can be said that the tool can only be used for simple, not extensive ER Models.

As ER modeling stays relevant, there are also contemporary approaches dealing with this model. However, most of these approaches don't deal with transforming ER models to relational ones. Examples include approaches for creating ER models from text using natural language processing [24]-[26] or applying ER modeling for creating specific models, e.g., for ontologies [27], Kanban systems [28] or software structures [29]. A small number of approaches deals with the relational transformation, but they either provide only very basic transformations with no practical application [30] or no novel transformations at all [31].

All in all, it can be stated that the proposed transformation approaches did not make it into applicable tools, mostly due to the lack of coverage of operational issues. On the other hand, modern editors seem to focus on simplified binary variants that are closer to relational tables than to Chen's ER model.

III. DEFINITION OF THE ER MODEL

As described in Sections I and II, there is a number of variations of ER models. However, in order to develop algorithms for transforming the ER model to the relational model, it is mandatory that modeling capabilities are known. Due to this, the ER model used in this work will be defined to ensure an unambiguous transformation.

When defining the modeling capabilities, one goal is to provide the modeler as much freedom as possible. The ER model defined in this paper is heavily based on the model presented by Kemper and Eickler [32]. This model includes the most prevalent modeling components such as entities, n-ary relations and attributes. It also contains existence-dependent types and covers generalization in form of IsA-Structures. The model is additionally extended by the attribute types ”Multi-valued attribute” and ”Compound attribute” according to Vossen [33].

An entity is the most basic ER component, covered within this paper. It does not have a direct connection with another entity, must have at least one identifying attribute and may be referenced by any number of attributes.

The relation is derived analogously to the entities from the basic principles of the ER model. Within this work, n-ary relations are supported. Every relation must connect at least two entity types and may be referenced by any number of attributes. To increase the modeling capabilities, the ER model also allows reflexive relations from one entity to itself.

The third component commonly used in ER models are attributes. In the context of this work, a distinction is made between three types of attributes. Regular single-valued attributes, multi-valued attributes and identifying attributes. To extend the capabilities of the model, regular and multi-valued attributes can be referenced by further regular and multi-valued attributes. This means that each regular and multi-valued attribute can act as a composite of other attributes. The attributes that form a composite are called composite attributes. Attributes, which are part of a composite attribute are also implicitly given the possibility to act as an composite attribute, which allows the multiple application of composition and multi-valuedness. With the multiple application of multi-valuedness and composition, complex attribute structures can be formed. These structures can form cyclic dependencies and ambiguities between attributes when they reference each other directly or indirectly via further attributes. Nevertheless, it must be guaranteed that each attribute can be uniquely assigned to exactly one entity or relationship. Each attribute can be assigned unambiguously to one entity or relationship if there is a direct connection or exactly one path to an entity or relationship via further attributes. The uniqueness requirement implies that attribute structures must have the form of trees with the corresponding entity or relation as root.

The transformation presented in this paper also supports existence-dependent types, which are also referred to as weak types. Weak types depend on a parent type for identification. In the ER model, this dependency is modeled by the use of a relationship between the parent entity and the existence-dependent entity, called a weak relation. It should be noted that in this ER model, the parent does not have to be a strong entity. Therefore, it is possible that a parent entity is also a weak entity, which in turn depends on another parent entity. Due to this multiple application of existence dependence, restrictions are required to uniquely assign a weak entity its dependent type. These are conceptionally analogous to the multiple application of the composition and multi-valuedness of attributes.

The ER model and transformation also incorporates generalization. Overall, generalization can be divided into a number of different types, characteristics, and constraints. In the context of this paper, generalization in the ER model is implemented exclusively by means of IsA structures, omitting the notation and transfer of specific properties of generalization. IsA structures associate multiple entities with each other. Each entity of the IsA structure acts as a subtype or supertype entity. An IsA structure references exactly one supertype entity and any number of subtype entities. The subtype entities of the IsA structure inherit all attributes of the supertype entity.

When defining the rules for IsA structures, a multiple-inheritance of the attributes of a supertype entity to a subtype entity must be excluded. To avoid this, a restriction to tree-structures, as with attributes, could be made but would be too restrictive. Instead the restriction is made based on the three following sets.

A) The ”subtype set” of an entity contains the entity itself and all entities, which inherit from the entity.
B) The ”supertype set” of an entity includes the entity itself and all other entities, which the entity inherits from.
C) The ”influenced type set” of an entity includes the subtype set and additionally for each entity in the subtype set the supertype set.

The sets defined above on the basis of an expression of several IsA structures are shown in Figure 1. The sets here start from the entity highlighted in blue. The entity within the area shown in red is part of the supertype set. The entities within the green area shown are part of the subtype set. The influenced type set contains all entities, which are highlighted in purple.

If an entity is connected as a subtype of an IsA structure the influenced type set of the entity must be disjoint with the supertype set of the supertype of the IsA structure. In contrast, when an entity is connected as the supertype of an IsA structure, the influenced type set of all subtypes of the IsA structure must be disjoint with the supertype set of the entity.

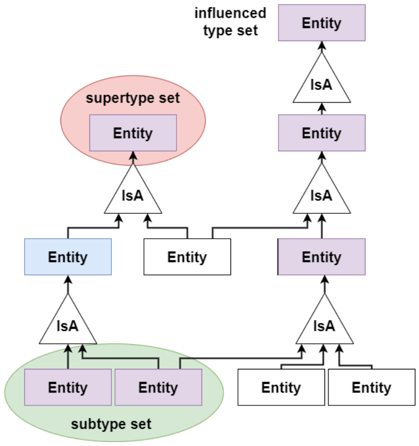


Fig. 1. Subtype-, supertype and influenced type set

In summary, the ER model discussed in this paper is extended by the following concepts:

- Multivalence of attributes

- Composition of attributes
- Reflexive, unary relationships
- N-ary relationships
- Existence dependency
- Generalization

IV. TRANSFORMATION

The implementable transformation of ER models is realized in several steps, which are executed sequentially.

A) Creating a data model structure of ER diagrams
B) Transformation of attributes
C) Transformation of IsA structures
D) Transformation of weak types
E) Transformation of relationships
F) Cascading of primary keys for attributes

These steps will be explained in more detail below.

A. Structural data model of ER models

In order to execute algorithmic approaches for translating the ER model into the relational model, a basic data model is required on which they can operate.

Any ER diagram essentially consists of elements such as entities, relationships, attributes, IsA structures, and associations between those elements. Therefore, the structure can be expressed directly as a graph. Furthermore, information about the cardinality between a relation and an entity can be stored in the edges of the graph. In case of a IsA structure, the edges also contain information about whether the connected node is a supertype or subtype.

In Section III, it was explained that attributes are always expressed in the form of trees. This makes it possible to further restrict the graph. Since all entities and relationships can have attributes, each of these elements acts as the root of a tree. Each attribute in the ER model is therefore represented as a node in the tree. The edges within the tree structure do not hold any additional information.

B. Transformation of attributes

The goal of this translation is to express any attribute structure by means of relations. From the previous subsection, it is known that all attributes are held within a tree. Therefore, only these trees have to be considered in this translation. If a tree consisting exclusively of single-valued attributes is considered, this can be solved trivially by creating a relation for the tree's root entity or relationship and adding each attribute, which is a leaf of the tree to that relation. Algorithmically, this can be done by traversing the tree in post-order and checking whether the current node is a leaf. However, if a tree contains multi-valued attributes, these cannot be added to the tree‘s root relation as a relation expresses a fixed sized schema and multi-valued attributes can take on any number of values. In this case, a standalone relation must be created for that attribute and a reference between it and the tree's root relation has to be created. In the case of composite attribute structures, attention must be paid between which relations a reference is created. It is important to note, that the referenced relation does not necessarily have to be the tree's root relation. As it is quite possible that the relation of a multi-valued attribute references a relation of another multi-valued attribute. The transformation of composite attribute structures can be executed by creating a relation for each attribute at the beginning. If the tree is then traversed in post-order, and the current node represents a multi-valued attribute a reference can be created between the current node‘s relation and the node‘s parent relation.

Listing 1. Transformation of attributes

```
1  Function TransformAttributeTree (Parent)
2      For Each Child in TreeNode //Execute post-order
3          TransformAttributeTree (Child)
4      End For
5      If TreeNode is Multivalued Attribute Then
6          TreeNode.Table <- marked
7      End If
8
9      If TreeNode is Leaf Then //Recursion resolution
10         Return
11     End If
12
13     For Each Child in TreeNode
14         If Child.Table is marked Then
15             If TreeNode.Children.Size = 1 Then
16                 //Handling of "forwarding" attributes
17                 MergeTable(TreeNode.Table, Child.Table)
18                 TreeNode.Table <- marked
19             Else
20               TreeNode.Table.References.Add(Child.Table)
21             End If
22         Else
23             MergeTable(TreeNode.Table, Child.Table)
24         End If
25     End For
26 End Function
27
28 Function MergeTable(ParentTable, ChildTable)
29     ParentTable.Columns.AddAll(ChildTable.Columns)
30     ParentTable.References.AddAll(ChildTable.References)
31     Delete ChildTable
32 End Function
```

For single-valued attributes, the relation can be merged with the relation of its parent node. Within the merge, all attributes and references of the child relation are transferred to the relation of the parent node. This procedure can be extended by skipping composite attributes, which consist of only one additional multi-valued attribute.

Listing 1 shows the algorithm for translating attributes in pseudo code. It is executed for each entity and each relation in the ER graph. The algorithm is explained below.

The initial situation of the algorithm, shown in Listing 1, is that the ER graph has been created. In addition, a relation is created for each attribute. As within this algorithm, relations which correspond to a single-valued attribute are successively unified. The relations which were created for multi-valued attributes are preserved. For these relations only the above mentioned references are created. In the shown algorithm the postorder traversal takes place in the lines 2 to 4, as well as 9 to 11. For the handling of multi-valued attributes, these are marked in each call in lines 5 to 7. This takes place before the recursion resolution, in order to seize also multi-valued attributes, which are leaves of the attribute tree. Otherwise, those would not be marked and would be merged in the following lines. In lines 13 to 25, each direct child attribute is handled for an attribute. If it is a marked attribute, a reference

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

to the child attribute is created in line 20. In the special case that the attribute has only one multi-valued attribute as a child, lines 15 to 18 are executed and a ”skipping” takes place, regardless of whether the current attribute is a multi-valued or single-valued attribute. If, on the other hand, it is a single-value attribute (line 23), the relation of the child attribute can be resolved by merging it with the relation of the current attribute. The merging itself is done by adding all columns and references of the child table to the parent table (lines 29 to 31).

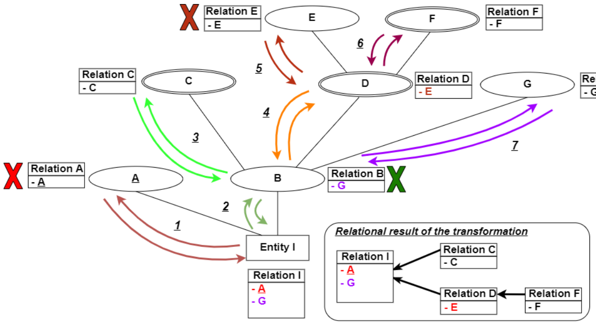


Fig. 2. Transformation of complex attribute structures

Figure 2 illustrates this algorithmic process. The arrows indicate, how the algorithm will process the attributes. The cross next to the relations shows that these have been resolved. The resolution takes place in the same call as the arrows shown in the same color. The result is shown on the bottom right. Note that the attribute value D consists here of many attribute values F and one attribute value E. A direct reference between the entity relation and the relation F could not represent this situation. Also note that the attribute B is merged within the algorithm. This is permissible because entity I is associated with exactly one value for the attribute G.

To complete the translation of the attributes into the relational model, the references between the created relations must be mapped in the form of foreign key dependencies. For this purpose, the remaining relations can be traversed in pre-order and primary keys can be added to the relations. These also act as foreign keys to the primary keys of the referenced relation. The cascading of the foreign keys is not executed directly, as it is only ensured that the root entity or relation contains all primary keys due to its identifying attributes. However, as new primary keys may be added if the entity is a weak entity or part of an IsA structure the immediate execution could lead to invalid references between the relations. Therefore, the execution of the cascading will take place at the end of the whole transformation process as Step F.

C. Transformation of IsA-Structures

The transformation of IsA structures is realized by means of foreign key dependencies between the subtypes and supertypes of the IsA structure. It should be noted that the relations of the entities must exist and the primary keys must be located in them. Because of this, the transformation of the attributes must take place before the transformation of IsA structures. Each subtype of an IsA structure inherits all primary keys of the supertype. In addition, these inherited primary keys refer to the upper type as foreign keys. If entities are part of several IsA structures, ”higher level” IsA structures have to be translated first to ensure that entities at lower levels receive all primary keys. To illustrate the translation order Figure 3 shows an entity-relationship model on the left and the relational model on the right. The red highlighted IsA structure can only be translated after the blue and green IsA structures. The blue one, on the other hand, only after the green one above it.

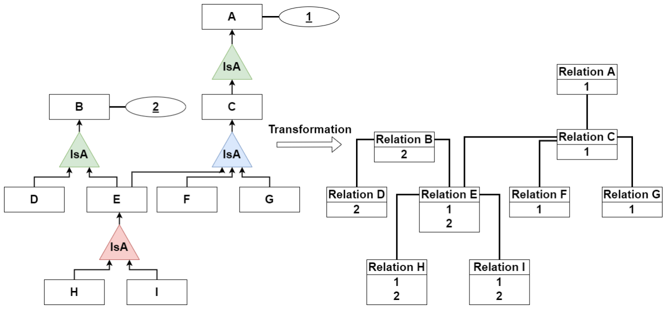


Fig. 3. Transformation order of IsA-Structures

Listing 2 shows the algorithm for transforming an IsA structure. Here, the processing order of the structures is maintained by traversing all IsA structures up to N times, where N is the number of IsA structures.

If the supertype of the selected IsA structure inherits from further IsA structures, and these have not yet been translated, the current pass is skipped (lines 6 to 10). Specifically, line 6 determines all IsA structures that must already be transformed in order to transform the current IsA structure. In Figure 3, this corresponds to the green and blue highlighted IsA structure in the case of the currently treated red IsA structure. Following this, line 7 checks whether all have already been transformed. If at least one IsA structure has not been transformed, the current call is skipped. This also applies if the selected IsA structure has already been translated (lines 2 to 4). If the IsA structure can be transformed in this call, the actual transformation takes place by creating the foreign key dependencies in lines 12 to 16.

Listing 2. Transformation of IsA-Structures

```
1   Function TransformIsAStructure(IsAStruct)
2     If IsAStruct is transformed Then
3       Return
4     End If
5
6     UpperLayerIsAs <- GetInheritedIsAs(IsAStruct.SuperType)
7     UnhandledUpperLayerIsAs <- UpperLayerIsAs !transformed
8     If UnhandledUpperLayerIsAs not empty Then
9       Return
10    End If
11
12    For Each SubType in IsAStruct.Subtypes
13      For Each PrimaryKey in SuperType
14        AddForeignKeyAsPrimaryKey(Supertype, Subtype)
15      End For
16    End For
17    IsAStruct <- isTransformed
18  End Function
```

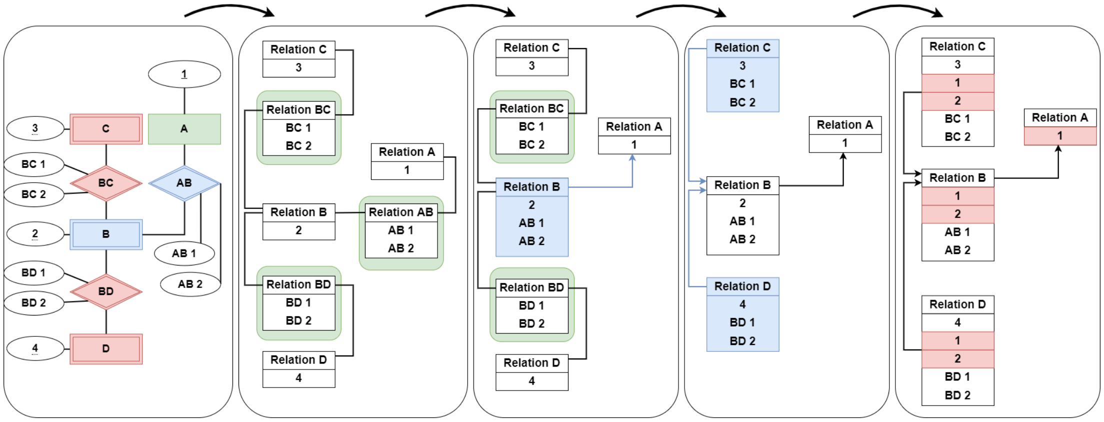


Fig. 4. Transformation of weak types

D. Transformation of weak types

For the translation of weak types to the relational model, it is a prerequisite that relations exist for all entities and relationships and that all attributes are already contained in them. Therefore, the transformation of attributes and IsA structures must be performed beforehand. IsA structures must be translated before, since a strong entity, on which a weak entity depends, can receive further primary keys during the translation of IsA-Structures.

Equivalent to the transformation of IsA-Structures, the translation order has to be considered. The translation has to start from weak entities, which have a connection to a strong entity or an already translated weak entity by means of a weak relationship. The relation of the weak relationship always has to be merged with the relation of the dependent entity. During the translation, the relation of the entity to be translated keeps a reference to the strong or already translated entity. This reference can then be used to create the foreign key dependencies.

The translation process of an ER diagram is shown in Figure 4. The figure starts immediately after the execution of the algorithms for the translation of attributes and IsA structures. The first rectangle shows an example ER model, which is transformed over a series of steps. The second step is the starting point of the algorithm, where all elements occur as a relation, conditioned by the previously executed attribute algorithm. The green highlighted elements represent the weak relationships in the ER model, which are required to be transformed. According to the mentioned translation order, the elements to be translated are determined and transformed in each step. In Figure 4, the blue elements are translated first, followed by the red elements.

Algorithmically, the merging of relations and reference creation from Figure 4 is shown in Listing 3. The compliance with the order is done, analogous to the translation of IsA structures, by checking all weak entities up to N times, where N equals the number of weak entities in the graph.

Listing 3. Transformation of weak types

```
1   Function TransformWeakEntity(WeakEnt)
2       If WeakEnt is transformed Then
3           Return
4       End If
5
6       WeakRelations <- GetConnectedWeakRelations(WeakEnt)
7       For Each WeakRel in WeakRelations
8
9           OtherEnt <- GetOtherEntity(WeakEnt, WeakRel)
10          If OtherEnt is no StrongEntity or
11                      is not transformed WeakEntity Then
12              Continue
13          End If
14          WeakEnt.Transformed <- true
15          WeakEnt.References.Add(OtherEntity)
16          MergeTables(WeakEnt, WeakRel)
17          Return
18      End For
19  End Function
```

In contrast to the IsA structure algorithm, the transformation algorithm resolves all connected weak relationships (line 6) and immediately tries to transform them (lines 7 to 18). If weak relationships are connected to a strong entity or an already transformed weak entity (lines 9 to 13), the current weak entity can determine its existence-dependent type and can be transformed (lines 15 to 16).

Note that the cardinalities of the weak relationship are not to be considered for the basic transformation, since these can only be 1:1 and N:1 towards the identifying type. The given algorithm realizes both functionalities by means of a foreign key dependency.

E. Transformation of relationships

Transformation of regular relationships requires prior execution of all previous algorithms, since all relations for entities, weak types and relationships require to have the complete primary keys.

Generally, there are three cases to consider when translating.

If a relationship connects two entities and the cardinality is N:M, the primary keys of the two entities are added to the relation of the relationship. These then reference the primary keys of the entity relations as foreign keys. The translation of N-ary relations is done regardless of their cardinality. The transformation of these is analogous to the translation of binary N:M relations. In this case, the relation of the relationship receives the primary keys of all connected entities. Each of this primary keys refers to the primary key of the corresponding entity in the form of a foreign key.

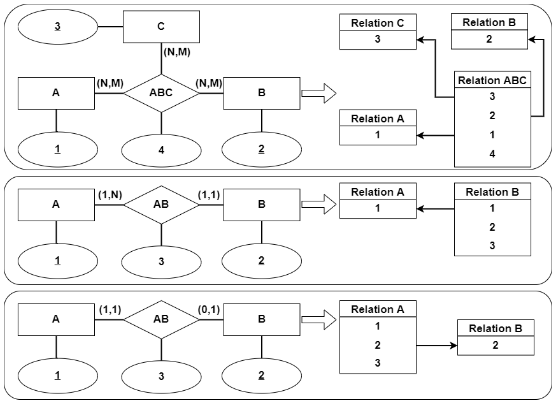


Fig. 5. Transformation of relationships

If the cardinality of the relationship is 1:N, the relationship is resolved by merging the relation of the relationship with the entities relation on the N side. In addition, this merged relation receives all primary keys of the opposite entity as normal attributes. These act as foreign keys on the opposite entity. In the third case, the relationship cardinality is 1:1. The translation is to be performed analogously to 1:N relationships, and the entity that receives the foreign keys and relation attributes must be specified for this purpose. Since this work, the Min-Max notation is used, the optionality has to be considered to avoid zero values. If one of the cardinalities describes an optionality, the attributes of the relationship and foreign keys are added to the entity on the other side. If both or none of the cardinalities describe an optionality, then the optionality is arbitrary.

Figure 5 shows the above cases. On the left side is the ER model. On the right side is the relational model resulting from the translation of the ER model. The first box in Figure 5 shows the transformation of N:M relationships, while the second displays a 1:N transformation. The last one shows the transformation of an optional 1:1 relationship.

V. PRACTICAL IMPLEMENTATION

Since the main focus of this paper is on practical application, we also provide an implementation of our concepts. To evaluate the algorithms given in this work a web application has been developed and put into operation, which implements all algorithms presented in this work. The application also contains a graphical editor.

Within the editor the user is able to create extensive and complex ER models, which are only restricted by a few limiting rules, required to allow an unambiguous logical assignment of the ER components. Based on the user-generated diagram, the presented algorithms are used to transform the diagram into a relational model without further user intervention.

To enforce correctly modeled ER diagrams, a validation process was implemented. This validation mechanism is designed in a user-friendly and proactive manner to ensure that a user is able to model fast and easy. Therefore, the validation offers maximum flexibility by only restricting actions which would necessarily lead to a violation of a rule and thus to an mandatory reverse action. Utilizing the presented algorithms and the validation procedure, the graphical editor is able to translate any model that can be modeled with it into the relational model. Furthermore, the editor visually presents the created relational model to the user.

To increase the practical applicability, an SQL generator was implemented within the application, which works on the basis of the generated relational model and generates SQL schema definitions. The generated SQL is in PostgreSQL dialect.

The graphical editor is shown in Figure 6. At first, a left side bar can be seen. Elements on this bar can be dragged into the drawing area to the right to create new elements. The use of drag & drop is intended to make it possible to create elements quickly and intuitively. The drawing area itself can be expanded endlessly to the right and bottom. If an element is selected, a side bar becomes visible on the right side, which provides additional options. These are, for example, assigning a name, deleting the element or creating a link to another element. In the case of Figure 6, a relation was selected which further enables the option to add new associations or edit the cardinalities to existing entities.

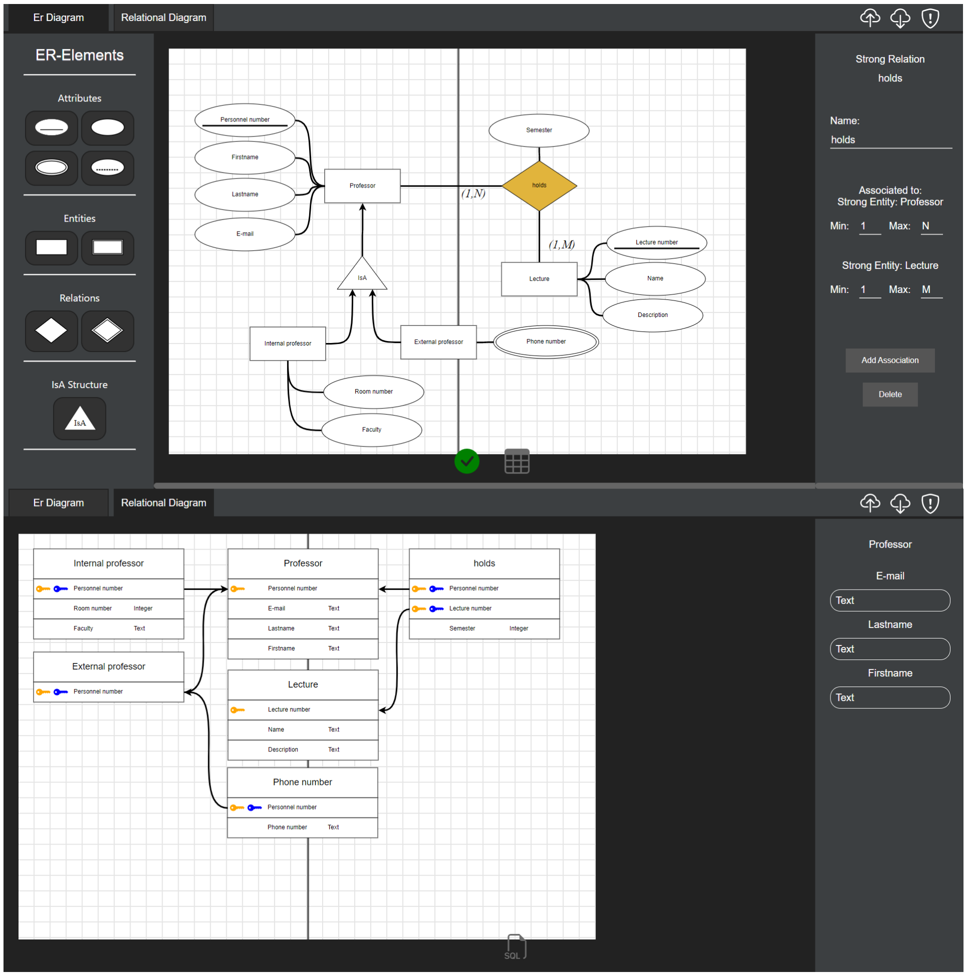
To further enhance the practical applicability, a save and load function has been implemented. Using this functionality by clicking buttons on the right top of the editor, a model can be saved in the form of a text file and at any time be loaded into the editor. By using the button at the center bottom the model can be transformed into the relational model, which will be, without the need of any further user interaction, transformed by the implemented algorithms and visually presented to the user. Furthermore, it is possible to freely switch between the conceptual and relational view using the tab bar in the upper left. In order to generate SQL code from the relational model, data types can be entered in the columns of the relational model.

The translation process from the ER model to the relational model has already been explained in detail. However, to make the transformation practically applicable and the editor effectively usable, two further building blocks have been integrated as mentioned before: The validation of the ER models created by the users and the automatic generation of SQL based on the generated relational model. Both of these will be described in the following.

A. Validation

An essential aspect for the practical applicability of the ER modeling tool, or ER modeling using a tool in general, is to ensure in a very user-friendly way that the ER model created within it complies with the definition for ER models given in Section III.

Several mechanisms to ensure compliance with the ER modeling rules have been explored. These range from native approaches such as holistic validation of the model at the time of translation, to approaches that take effect within the



Strong Relation
holds

Name: holds

Associated to:
Strong Entity: Professor
Min: 1 Max: N

Strong Entity: Lecture
Min: 1 Max: M

Internal professor	
Personnel number	
Room number	Integer
Faculty	Text

External professor
Personnel number

Professor	
Personnel number	
E-mail	Text
Lastname	Text
Firstname	Text

Lecture	
Lecture number	
Name	Text
Description	Text

Phone number	
Personnel number	
Phone number	Text

holds	
Personnel number	
Lecture number	
Semester	Integer

Professor
E-mail
Text
Lastname
Text
Firstname
Text

Fig. 6. ER Modeling Tool

modeling process and restrict user options in advance that could lead to an invalid ER Model.

The approach most suitable for this tool is briefly presented below. The approach can be described as an extended variant of the Correctness by Construction principle [34].

The basic idea of this validation system is to proactively prevent actions within the modeling process that would inevitably lead to a violation of a rule. In contrast to a pure Correctness by Construction approach, the idea is extended by not only allowing actions that transition the ER model from one valid state to another, like adding a new entity to the model. The extension involves the introduction of an additional state to allow the user much more flexibility during the modeling process.

To understand exactly what restrictions would be involved if the additional state is not used, an example modeling process is shown in Figure 7.

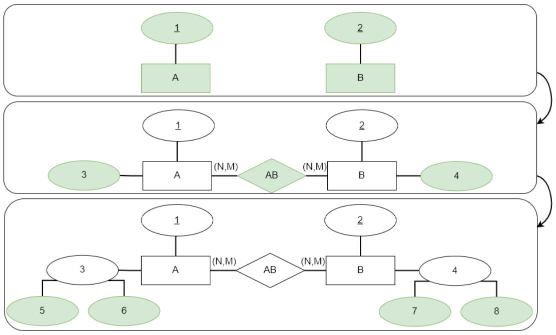


Fig. 7. Modeling process with simple valid to valid state transitions

The boxes shown in Figure 7 show the gradual creation of the ER model across the valid states. The elements shown in green are the newly added ones. It shows up clearly that if every action of the user would be checked for compliance with the rules, the user would be strongly limited in his possibilities. For example, it would not possible to freely delete elements or to model attributes first.

Modeling using the new state allows these restrictions to be avoided.

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

The additional state in which the ER model can be is defined as follows: An entity-relationship model is in a partially consistent state if it can be transformed by insertion operations, consisting of adding elements and associations, into a state that satisfies all ER rules defined in Section III.

This definition of a state of an ER diagram is a subset of the possible valid and invalid forms of an ER diagram. The state implicitly assures through its definition that it can be transformed into a valid ER model, but still gives the modeler a lot of freedom.

For example, modeling using this state does not restrict the insertion of elements, since ER models are generally not finite and each element can be integrated into an ER model at any time. In addition, deleting elements is not restricted in any way. If the original state of the ER diagram is in a partially consistent state, all ER diagrams that are changed by deletion operations are also in a partially consistent state, since they can be restored to their original state by a reverse insert operation.

This state practically only restricts operations that would force the ER model into an invalid state, such as an association of two entities without a relationship. In order to restore this model to a valid state, it would be necessary to remove exactly this association. Using this state, the modeler is given the highest possible freedom in modeling without compromising the consistency of the ER model.

In practice, the check is performed with every action of the modeler. For example, if an element is selected, all other elements of the model are checked one after the other. During the check, an association is created between one of the elements and the selected element and it is examined whether the model still corresponds to the new state. If this is the case, an association between these two elements is permitted. In Figure 6, this would be indicated by the permissible elements being highlighted.

Visually, the gradual creation of the er model with this state is exemplified in Figure 8. The boxes and elements highlighted in green can be interpreted in the same way as in Figure 7, where the elements shown in green are the newly created ones.

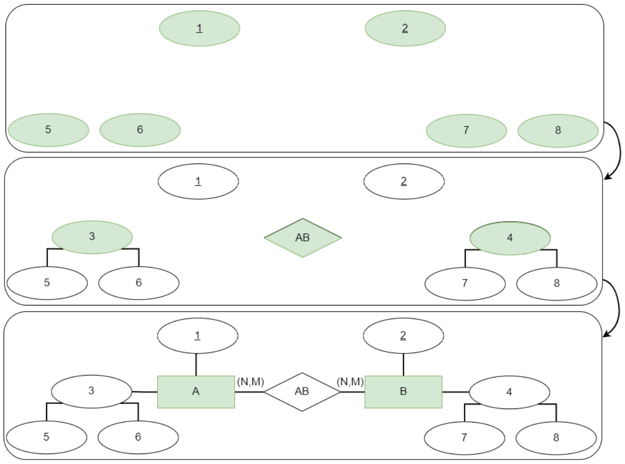


Fig. 8. Modeling process with transitions to additional state

However, Figure 8 also shows that the ER diagram can be far from a valid state. Therefore, a feedback system was installed, which informs the user at which parts the model is not yet fully valid by checking the current state of the er model against a valid one.

B. SQL Generation

This section provides a brief summary of the feature for generating SQL from the relational model. In principle, the relational model generated from the ER model can be represented as a graph in which tables form the nodes and foreign key constraints form the edges between the nodes. An example is shown in Figure 9. On the left side a relational model is shown. The right side, in turn, shows the resulting directed graph.

A topological sort is performed on these graphs. Thus, the result of the algorithm is an ordered sequence of tables, where each table occurs before it is referenced through foreign key constraints of other tables. SQL code can then be generated for each table using an SQL template. These statements are combined afterwards to achieve an SQL script for the whole model. This allows a sequence of SQL statements to be generated for a relational model, which can be executed immediately one after the other, thus enabling the modeler to implement the relational model in a target database.

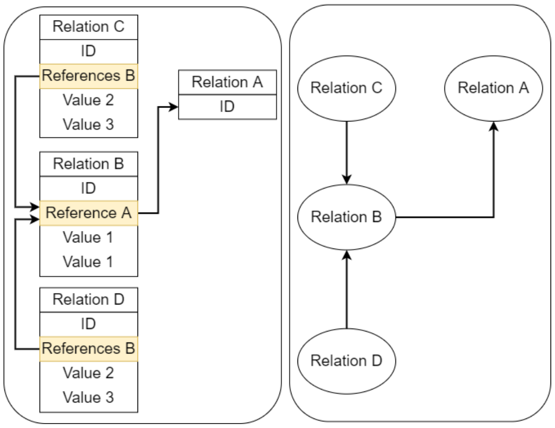


Fig. 9. Relational model as graph

VI. EVALUATION

In this paper we proposed a transformation of ER models to relational models with focus on practical applicability. Therefore, we also chose a practical kind of evaluation, which is two-fold.

On the one hand, we implemented the editor tool described in the previous section. In its current state, the editor allows for ER modeling and an automated transformation of all mentioned concepts. By means of these results, it is shown that the given algorithms are implementable and are capable of performing a transformation of ER models into relational models. Using this editor, we also conducted a preliminary evaluation regarding the correctness of the transformation by testing the editor with a predefined set of ER models containing different combinations of modeling elements. The evaluation was successful as the editor correctly transformed all supported concepts. To enable a broader evaluation and application of the editor, we made it available open source [35].

On the other hand, to gather empirical evidence about the capability of the transformation and the functionality of the editor, we conducted a practical evaluation with 15 undergraduate computer science students of the third semester. All of the subjects attended the lecture ”Database Systems”, five of them had prior database knowledge. For the evaluation, they had to create two ER models from a textual description. Both of these were also part of the lecture and the students had already completed the tasks two months before as exercises with an editor of their choice. That way, we could also obtain comparative information regarding our editor and others. The modeling tasks included all concepts relevant for the database lecture: Entities, different types of relationships, different types of attributes, and inheritance. After completion of the tasks, the subjects had to fill out a questionnaire. The results will be discussed in the following while Appendix A contains several figures summarizing the results.

The first question dealt with the intuitiveness of the editor. The subjects had to answer with a score from 1 (very bad) to 5 (very good). The average of the answers (cf. Figure 10) of the 15 subjects was a score of 4.13. The subjects also stated that none of the necessary modeling elements were missing (cf. Figure 11). Optional comments regarding missing features of the editor targeted usability improvements (multi selection of elements and shortcuts for commands). However, the provided modeling elements did mostly behave as the subjects expected (cf. Figure 12). There were also optional comments about behavior of the editor/elements that should be improved:

- Editing pane too small on small displays 1
- No non-integer primary keys 1
- Resizing of entities, relationships, and attributes 1
- N-ary relationships 1
- Creating relationships is cumbersome 1

The subjects were also asked about their previous editor of choice and provided the following answers:

- Enterprise Architect 7
- Draw.io 3
- ERDplus 1
- None 1
- Metatron 1
- Miro 1
- Visio 1

Comparing the modeling experience of the other editor to the one presented in this paper, the majority of the subjects preferred the latter (cf. Figure 13). In addition, most of the subjects would prefer our editor or a combination in the future (cf. Figure 14). When asked for the advantages of our editor, the subjects provided the following answers:

- Conversion to database schema 5
- Simplicity 5
- Intuitiveness 4
- Modeling support 2
- Contains all necessary elements 2
- Quicker modeling 1

The subjects also provided information about disadvantages of our editor compared to the prevalent ones:

- GUI visual representation 6
- Editing of nodes cumbersome 1
- No automatic saving 1
- Draw area too small 1

Besides the editor, the evaluation also focused on the implemented transformations to the relational model and to SQL. Nearly all of the subjects stated that the transformation to the relational model worked (cf. Figure 15). However, it turned out that the problems two subjects had with the transformation were caused by a visual problem: The current state of the editor only supports one browser. If executed in another, the relational visualization has some issues. This was also stated by the users when asked for problems regarding the transformation:

- Visualization problem 4
- Data types missing in relational model 1

Due to all subjects, the transformation to SQL worked well (cf. Figure 16). The same applied for the correctness of the created models (cf. Figure 17). When comparing the transformation of the editor to their previously created own manual transformation nearly all subjects rated the one of the editor equally well or better (cf. Figure 18).

In summary it can be stated that the implemented transformations worked nearly perfect for all of the subjects in this practical evaluation. The editor software was also perceived rather positive. However, for the latter there was also constructive criticism, mainly relating to usability. We will take this into account and use it for improving the editor in future versions.

VII. CONCLUSION

Despite its age, ER modeling is still the most prevalent way of creating conceptual data base schemata. Since its advent in the 1970s, various extensions have been proposed. Due to this, many different flavors are currently used in modern editors. To be applicable as technical database schema, the ER models have to be transformed into relational models. This can be a complicated and error-prone task. Therefore, various standardized transformation approaches have been proposed over the decades. However, these approaches remained rather theoretic and did not include operational semantics. Thus, no tool support was established utilizing them and the transformation process remained manually to a large extend.

Despite this issue, modeling support for ER models was achieved. To date, a high number of editors is available in different flavors. There are diagram tools offering ER diagram creation, database clients with ER schema creation options, or other editors like UML editors incorporating database modeling. While some of them only provide diagrams, others enable the direct application to relational databases. However, there is still no practical automated transformation of ER diagrams to relational ones. Editors offering this do not enable the modeling of real ER diagrams but rather enhanced DB diagrams or omit important prevalent concepts like n-ary relationships or generalization.

To tackle this issue, we proposed an approach for transforming ER models to relational ones with a strong focus on applicability and operational issues. The ER model incorporates the most prevalent and necessary concepts [32] as n-ary relationships, multi-valued and composite attributes, or generalization. All of these can be correctly transformed in any meaningful combination enabling great flexibility for the input models.

To prove the applicability of the proposed approach, we have implemented a graphical ER editor capable of creating diagrams containing all mentioned concepts as well as a an automated transformation to relational models. As practical applicability was our focus, we also added a validation mechanism to the editor that guarantees the creation of correct and transformable ER models while providing the user as much flexibility as possible and a good user experience. Furthermore, we have conducted an evaluation study with 15 subjects investigating the correctness and applicability of our approach as well as the usability of the proposed editor. The findings of this study include that the approach works well and is usable within the created editor. Comments for improvement targeted only usability issues of the editor. The editor is publicly available for further testing and improvement [35]. All in all we have shown a transformation approach that can easily be implemented in editors. This can aid future conceptual database modeling, spare time, and reduce errors resulting from manual transformation.

Our future work will focus on promoting the transformation approach and the editor. We will use the findings of our study to improve the usability of the editor and make it appealing to larger user numbers. This enables future evaluations with broader audiences. Further, we will add additional features to the approach as well as the editor. This includes additional and optional concepts to the ER models, like composition structures as well as other diagram types and transformations.

REFERENCES

[1] G. Grambow and S. Ruttman, "A Practical Automated Transformation of Entity Relationship Models to Relational Models," 15th Int'l Conf on Advances in Databases, Knowledge, and Data Applications, pp. 5–12, 2023.

[2] P. Chen, "The entity-relationship model—toward a unified view of data," ACM Trans. on DB Sys. vol. 1, no. 1, pp. 9–36, 1976.

[3] R. Elmasri and S.B. Navathe, Fundamentals of Database Systems. Benjamin/Cummings, Redwood City, 1994.

[4] S. Spaccapietra and C. Parent, "ERC + : an object based entity-relationship approach," Conceptual Modeling DBs and Case: An Integrated View of Information Systems Development, John Wiley, 1992.

[5] B. Thalheim, "Extending the entity-relationship model for a high level, theory-based database design," 1st Int. East West Database Workshop, pp. 161–184, 1990.

[6] T.J. Teorey, "Database Modeling and Design," The Entity-Relationship Approach, Morgan-Kaufmann, San Francisco, 1990.

[7] J. Makowsky, V. Markowitz, and N. Rotics, "Entity-relationship consistency for relational schemas," Int. Conf. on Database Theory, pp. 306–322, 1986.

[8] A. D'Atri and D, Saccà, "Equivalence and Mapping of Database Schemes," VLDB '84, pp. 187–195, 1984.

[9] E. Wong and R. Katz, "Logical design and schema conversion for relational and DBTG databases," Int. Conf. on ER Approach to Sys. Analysis and Design, pp. 311–321, 1979.

[10] S. Jajodia, P.A. Ng, and F.N. Springsteel, "The Problem of Equivalence for Entity-Relationship Diagrams," IEEE Trans. on SE vol. 9, no. 5, pp. 617–630, 1983.

[11] T. Teorey, D. Yang, and J. Fry, "A logical design methodology for relational databases using the extended entity-relationship model," ACM Computing Surveys (CSUR) vol. 18, no. 2, pp. 197–222, 1986.

[12] T. Ling, "A Normal Form For Entity-Relationship Diagrams," 4th Int. Conf. on the ER approach, pp. 24–35, 1985.

[13] H. Sakai, "Entity-relationship approach to the conceptual schema design," SIGMOD '80, pp. 1–8, 1980.

[14] Lucidchart. Last visited: 2024.05.14. [Online]. Available: https://www.lucidchart.com

[15] Draw.io. Last visited: 2024.05.14. [Online]. Available: https://app.diagrams.net/

[16] VisualParadigm. Last visited: 2024.05.14. [Online]. Available: https://online.visual-paradigm.com

[17] ERDPlus. Last visited: 2024.05.14. [Online]. Available: https://erdplus.com/

[18] pgAdmin. Last visited: 2024.05.14. [Online]. Available: https://www.pgadmin.org/

[19] MySQL Workbench. Last visited: 2024.05.14. [Online]. Available: https://www.mysql.com/products/workbench/

[20] Outsystems. Last visited: 2024.05.18. [Online]. Available: https://www.outsystems.com/

[21] Mendix. Last visited: 2024.05.18. [Online]. Available: https://www.mendix.com/

[22] Enterprise Architect. Last visited: 2024.05.14. [Online]. Available: https://www.sparxsystems.de/

[23] C. Fahrner and G. Vossen, "A survey of database design transformations based on the entity-relationship model," Data and Knowledge Engineering vol. 15, no. 3, pp. 213–250, 1995.

[24] M. Kasra Habib, "On the Automated Entity-Relationship and Schema Design by Natural Language Processing," Int. J. of Engineering and Science vol. 8, no. 11, pp. 42–48, 2019.

[25] P. G. T. H. Kashmira and S. Sumathipala, "Generating Entity Relationship Diagram from Requirement Specification based on NLP," 3rd Int. Conf. on Information Technology Research, pp. 1–4, 2018.

[26] S. Ghosh, P. Mukherjee, B. Chakraborty, and R. Bashar, "Automated Generation of E-R Diagram from a Given Text in Natural Language," Int. Conf. on Machine Learning and Data Engineering, pp. 91–96, 2018.

[27] M. Ahsan Raza, M. Rahmah, S. Raza, A. Noraziah, and R. Abd. Hamid, "A Methodology for Engineering Domain Ontology using Entity Relationship Model," Int. J. of Advanced Computer Science and Applications vol. 10, no. 8, pp. 326–332, 2019.

[28] K. Ĺachová and P. Trebuňa, "Modelling of Electronic Kanban System by Using of Entity Relationship Diagrams," Int. Scientific Journal about Logistics vol. 6, no. 3, pp. 63–66, 2019.

[29] A. Ramírez-Noriega, Y. Martínez-Ramírez, J. Chávez Lizárraga, K. Vázquez Niebla, and J. Soto, "A software tool to generate a Model-View-Controller architecture based on the Entity-Relationship Model," 8th Int. Conf. in Software Engineering Research and Innovation, pp. 57–63, 2020.

[30] Y. Liu, X. Zeng, K. Zhang, and Y. Zou, "Transforming Entity-Relationship Diagrams to Relational Schemas Using a Graph Grammar Formalism," IEEE Int. Conf. on Progress in Informatics and Computing, pp. 327–331, 2018.

[31] L. Yang and L. Cao, "The Effect of MySQL Workbench in Teaching Entity-Relationship Diagram (ERD) to Relational Schema Mapping" Int. J. Modern Education and Computer Science vol. 7, pp. 1–12, 2016.

[32] A. Kemper and A. Eickler, Datenbanksysteme. Eine Einführung [English: DBMS. An Introduction] Oldenbourg, 2015.

[33] G. Vossen, Datenmodelle, Datenbanksprachen und Datenbankmanagementsysteme [English: Data Models, Database Languages and DBMS] 5. Aufl. Oldenbourg, 2005.

[34] T. Bordis, T. Runge, A. Kittelmann, and I. Schaefer, ”Correctness-by-Construction: An Overview of the CorC Ecosystem,” ACM SIGAda Ada Letters vol. 42, no. 2, pp. 75–78, 2023.
[35] Editor Implementation. Last visited: 2024.05.14. [Online]. Available: https://github.com/SimonRuttmann/ERModellingTool

APPENDIX A. STUDY RESULTS

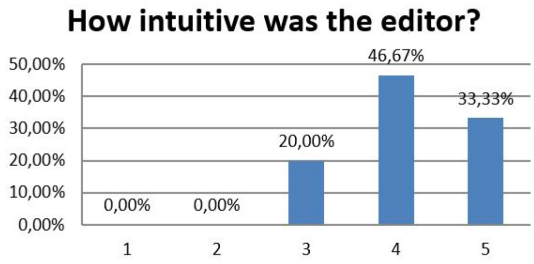


Fig. 10. Intuitiveness of the editor

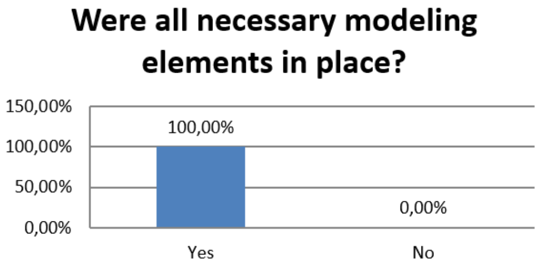


Fig. 11. Modeling elements in place

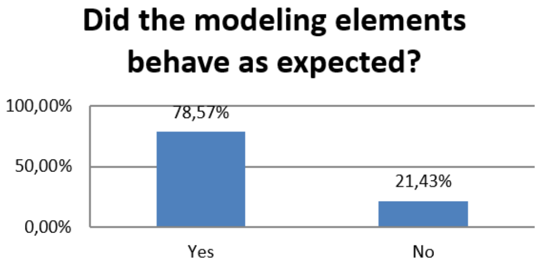


Fig. 12. Elements behave as expected

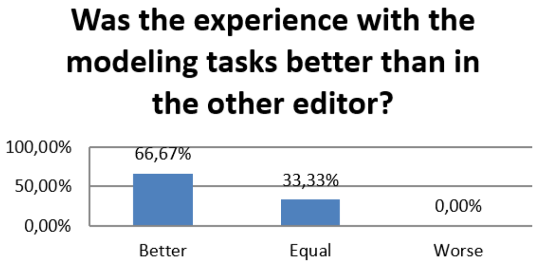


Fig. 13. Modeling experience

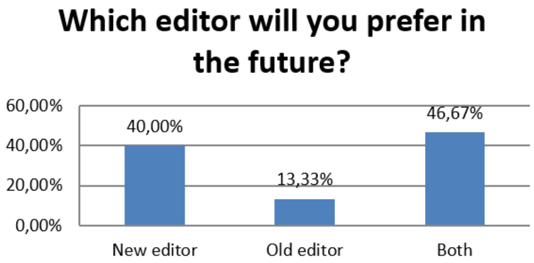


Fig. 14. Editor preference

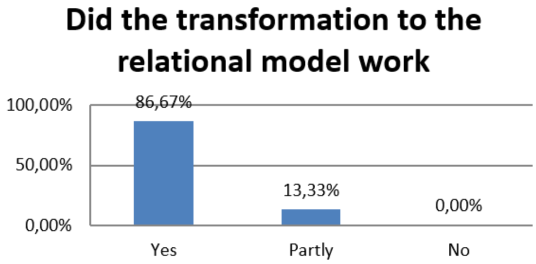


Fig. 15. Transformation to relational model

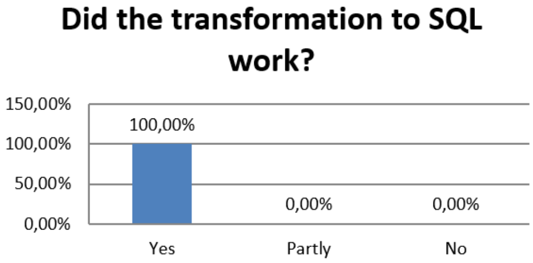


Fig. 16. Transformation to SQL

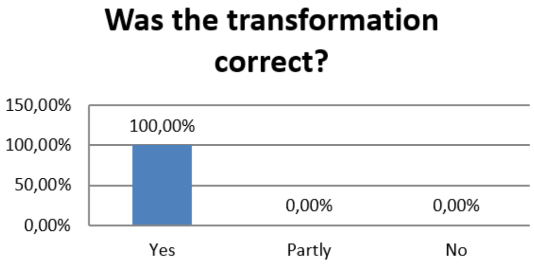


Fig. 17. Transformation correctness

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

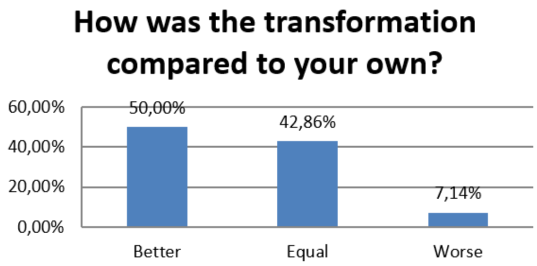


Fig. 18. Transformation comparison

Attempt to Design Electronic Bulletin Board Based on Information-Sharing Behavior Observations with Analog Bulletin Boards

Kanayo Ogura
Faculty of Software and Information Science
Iwate Prefectural University
Takizawa, Iwate, Japan
e-mail: ogura_k@iwate-pu.ac.jp

Ryotaro Hoshi
QUICK CO., LTD.
Tokyo, Japan
e-mail: hoshi-ryotaro@919.jp

Abstract—**In this study, based on the hypothesis that incorporating an analog information sharing environment, such as a station bulletin board, into a digital information sharing environment would enable casual and informal information sharing across small communities within a large organization, we conducted about 2 months of observation of whiteboard usage behaviors. Based on the results, we designed and implemented an electronic bulletin board through a large display that has functions to promote messages, improve their quality, and process them according to their importance, and allows users to post and view messages while conversing. We then conducted a user study for about 5 months to evaluate the effectiveness of the features of our electronic bulletin board based on the findings from our whiteboard observations and to test our hypothesis. As a result, we found that there were an average of 3 to 10 posts per day, including new posts and replies. In terms of reading, the survey found that about 90% of the 307 respondents (about 15% of all students) had read the bulletin board, confirming that our electronic bulletin board is being used daily as an information sharing tool. In addition, the "I saw it" reaction button was used for about 30% of all new posts and replies (552 posts), and the "Delete it" button was used for about 10%. Analysis of the message content in conjunction with the reaction buttons showed that they contribute to maintaining the quality of posts and judging the importance of posts, confirming the effectiveness of the implemented functions. Furthermore, the survey found multiple responses indicating that information sharing across departmental boundaries was possible, confirming that the hypothesis that by incorporating an analog information sharing environment, such as a station bulletin board, into a digital information sharing environment, it would be possible to achieve casual and informal information sharing across small communities within a large organization, is correct.**

Keywords-component; informal information sharing; electronic bulletin board; analog bulletin board; user study.

I. INTRODUCTION

This study is aimed at providing an unconventional information-sharing platform to make the useful information in casual conversations with people within large organizations such as universities and companies. In our previous study [1], we set up an analog information sharing space using a whiteboard in our university, observed users' writing and browsing behaviors, and examined the elements necessary to promote lightweight and informal information sharing to clarify the effective aspects of information sharing in the analog world. On the basis of observations of the whiteboard, we designed and implemented an electronic bulletin board for information sharing and confirmed through trial experiments that users continuously posted and replied to messages on the board.

In this paper, we conduct a more detailed analysis of the observed information sharing behavior in the analog world and a more detailed analysis of the user study of the electronic bulletin board we designed and implemented. In addition, we verify the effectiveness of the various functions of the implemented bulletin board, and examine whether the ultimate goal of this research—enabling casual and informal information sharing that transcends the boundaries of small communities within large organizations—is achievable or not.

In any organization, there are likely people who have already solved a particular problem or answered a particular question. Informal communications with such people can often contain useful information. However, information sharing that leads to problem-solving is often limited to close relationships. In large organizations such as universities and companies, people typically spend most of their time within small communities such as research laboratories or departments. Consequently, opportunities for information sharing beyond the boundaries of these small communities are limited. In some organizations, team communication services or social networking services (SNS) are used as a means of digital information sharing at the research laboratory or department level to support organizational information sharing [2] [3]. However, superiors and supervisors often use these platforms as a one-way communication channel. Community members perceive these platforms as formal spaces, hindering casual and informal information sharing.

To explore informal information sharing beyond the boundaries of existing organizations in the digital world, we consider the informal information exchange and sharing opportunities we experienced in the past through the use of online collaborative document-editing tools in a context-free and free-flowing manner. To further explore informal information sharing environments similar to the one we experienced, we also consider the analog information sharing environment of bulletin boards that were once installed in railway stations in Japan. These boards were originally intended for leaving messages about meeting places, such as for people who were late for appointments. However, they were often used for other purposes, such as sharing travel impressions, looking for friends, or scribbling graffiti. We believe that by incorporating an environment similar to these station message boards, where users can freely write in a free context, into digital information sharing environments, it is possible to promote informal and casual information sharing across small communities within large organizations.

Based on the above discussion, this study hypothesizes that by incorporating analog information sharing environments like the bulletin boards at train stations into digital information sharing environments, it will be possible to facilitate informal and casual information sharing across small communities within large organizations. To test this hypothesis, we set up a whiteboard in our university as an information sharing space where people can freely write whatever they like. We observed and analyzed the users' writing and browsing behavior to examine which elements of the analog information sharing environment should be incorporated into the digital information sharing environment. Based on the results, we designed and implemented an electronic bulletin board as a digital information sharing environment. We then conducted a user study of the implemented electronic bulletin board to evaluate its effectiveness and usefulness.

This paper is organized as follows. Section II reviews related research. Section III describes the observational study of the installed whiteboards. Section IV presents the design and implementation of the electronic bulletin board based on the results of the previous section. Section V describes the user study of this electronic bulletin board. Section VI discusses the findings and limitations of the study, and Section VII concludes this paper.

II. RELATED WORK

While there has been extensive research on the design and development of information sharing environments, recent trends indicate a growing reliance on existing team communication services and social networking services (SNS) for digital information sharing [2] [3]. Existing research in this area primarily focuses on modeling user relationships and posting activities on online forums [4] or exploring asynchronous online information sharing as a potential solution to collaboration fatigue in remote meetings [5]. To the best of our knowledge, there has been very little research on the design and development of informal information sharing environments using large displays. This paper provides an overview of several studies on information sharing that is achieved through the installation of large displays in shared spaces, similar to our study.

Nishimoto et al. [6] conducted an important study on promoting synchronous information sharing in large-scale organizations. In their system, a person with a transponder, a device that automatically sends a signal when it receives a different signal, approaches a large display in a shared space, and a question registered in advance by the person is displayed. This facilitates synchronous information sharing with users of the shared space who happen to see and discuss the question. The advantage of their system is that it does not require information providers to register their information with the system in advance, while general knowledge-management software requires users to register their information with the system. However, this approach focuses on synchronous information sharing and has the drawback that users cannot re-read previously displayed questions. In this study, we propose an electronic bulletin board that supports both synchronous and asynchronous information sharing. Users can engage in synchronous information sharing by directly conversing with each other next to the bulletin board, and they can also share information asynchronously by replying to previously posted content. However, one issue to consider is how long to keep posts displayed. To address this issue, we plan to conduct observational studies of posting and reading behavior on whiteboards and design the electronic bulletin board based on the observation results.

Snowdon et al. [7] proposed a recommendation system that semi-automatically displays filtered information on the basis of user comments and feedback for each post. One of the features of their system is that it gives users a more organic impression by randomly arranging the posted information when it is displayed. Their system's advantages include the random arrangement of post information to give users a more organic impression, which we also intend to incorporate into this study. On the other hand, when they actually operated their system, they found that the contents of the posts were often suited to the characteristics of the organization, but there was a problem that the posts were biased towards some users. To address this drawback, we intend to incorporate the findings of Fortin et al. [8] that suggest that displaying available posting spaces while maintaining a well-utilized appearance can attract users.

Chiba et al. [9] proposed “Attractiblog,” a system that displays company blogs on a large display in a communication space to induce face-to-face informal communication to support information sharing in large organizations. This system uses ID tags to identify users and can display blog posts that are relevant to the situation, such as those commented on by users who are currently in the communication space. This system has the advantage of using company blog posts, which can facilitate discussions about the organization and lead to more meaningful information sharing. However, this system has the drawback of requiring real-name use. This is because they also presented the results of a survey [10] that showed that anxiety about the possibility of providing incorrect information is an obstacle to knowledge sharing in workplace information sharing, such as on company blogs. There are also reports that self-esteem is higher when interacting anonymously through system-mediated communication [11], leading to more voluntary self-disclosure [12]. Therefore, we assume anonymous use of our bulletin board.

Greenberg et al. [13] proposed “The Notification Collage,” is used to share information not with individuals but with the community as a whole by allowing people in a small community to post their daily discoveries to the system. This system offers several advantages over conventional chat tools. Firstly, posted information is not displayed in a linear list, but rather randomly arranged like an analog bulletin board. This creates a more organic and visually appealing experience. Secondly, new posts are stacked on top of older ones, ensuring that the latest information is always visible to users, similar to chat tools. In this study, we intend to incorporate information presentation methods that correspond to information freshness into the design of the electronic bulletin board, including random arrangement and keeping the latest information visible. On the other hand, the Notification Collage was designed to be used on both large displays and personal devices. However, it was found that most users preferred to post and view information from their personal devices, resulting in underutilization of the large displays and limited opportunities for synchronous information sharing. This can be considered a drawback of the system. As we believe that synchronous information sharing is also important, this study focuses solely on the use of large displays.

III. OBSERVATIONAL STUDY

In this study, we set up two whiteboards in our university for about 2 months and observed posting and browsing behavior to realize lightweight and informal information sharing in the digital world that incorporates effective elements of information sharing in the analog world. The observation aimed to elucidate the elements necessary for beneficial information sharing on whiteboards, where people can freely write whatever they want, and to examine the elements necessary for facilitating information sharing on an electronic bulletin board.

A. Overview of Observations

We installed two whiteboards in the corridors on the 2nd and 3rd floors of our faculty building at our university (see Figure 1) for about 2 months. The reason is that these locations are conspicuous to students on their way to the student hall building, where the cafeteria and store are located, and to the common lecture building, where many lectures are held.

We set up a pen and a cleaner on each whiteboard (Figure 2). To create an environment that encourages student interaction through the whiteboards, we posted a sign next to each whiteboard explaining that "this whiteboard is a space for writing questions and answers about student life."

To collect data to be used to analyze information-sharing behavior, we took pictures of the entries on each whiteboard every weekday evening. From these pictures, we manually transcribed the contents written on each whiteboard and organized the contents of the writings and their relationships with other writings (related topics, question-answer relationships, etc.).

Furthermore, we conducted a questionnaire survey on the use of these whiteboards and information sharing related to daily life for all students in our university after the observation period. Additionally, we displayed a notice stating that the content written on the whiteboard would only be used for research purposes. We also informed participants in advance that if they could not consent to having their intended whiteboard writings utilized, they should refrain from writing on the whiteboard.

The observation period was approximately 2 months and was conducted on weekdays, excluding weekends and public holidays. To investigate the necessary requirements and incentive mechanisms for designing electronic bulletin boards, the observation period was divided into 5 periods, and the conditions for the whiteboards were changed on the 2nd and 3rd floors during each period. The 5 periods and conditions are as follows. Note that the 4th, 5th, and 7th weeks were excluded from the target period because they included weekdays with public holidays.

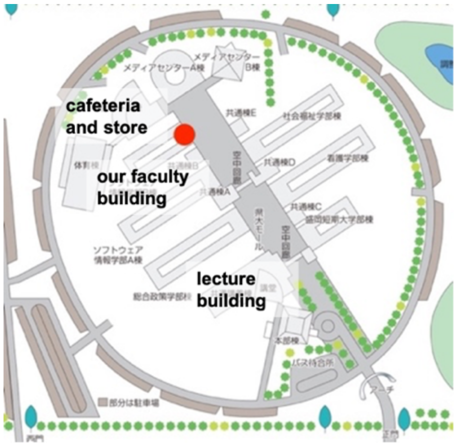


Figure 1. Campus plan of our university and locations of whiteboards [Red circles indicate locations (2nd and 3rd floor)].

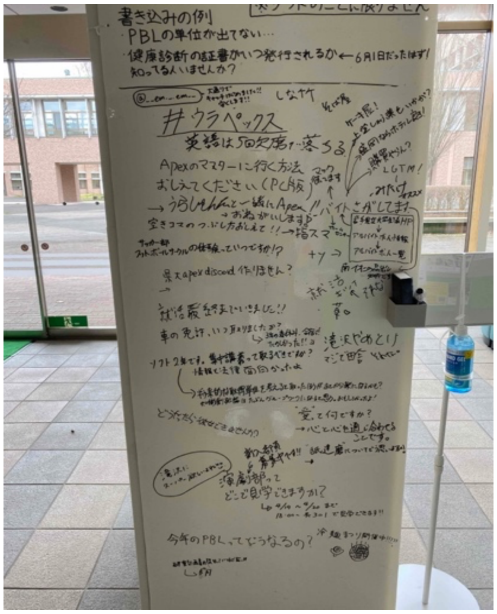


Figure 2. Installed whiteboard

1) Period 1 (1st week)

The purpose was to investigate whether spontaneous posts would occur on new whiteboards without specific instructions. The observation started with the 2nd and 3rd-floor whiteboards in a blank state.

2) Period 2 (2nd week)

The purpose was to investigate whether spontaneous deletion would occur when there was no new writing space on the whiteboards and to investigate the effects of keeping posts. The conditions were as follows:

- 2nd floor: The observation started with some of the remaining posts from Period 1. We would delete the posts if two days had passed since the last post or the whiteboard was 80% filled.
- 3rd floor: The observation started with the remaining posts from Period 1. We would not perform any regular deletion of posts.

3) Period 3 (3rd week)

The purpose was the same as that of Period 2. The conditions were as follows:

- 2nd floor: We regularly deleted posts following the same rules as Period 2.
- 3rd floor: The observation started with all of the posts from Period 2 deleted. We would not perform any periodic deletions.

4) Period 4 (6th week)

The purpose was to investigate whether the number of questions would increase. The conditions were as follows:

- 2nd floor: We would periodically delete posts. We also instructed participants to add a “Q” mark at the beginning of any question posts.
- 3rd floor: We would periodically delete posts.

5) Period 5 (8th week)

The purpose was to investigate whether the presence or absence of dividing lines would affect the number of posts. The conditions were as follows:

- 2nd floor: We would periodically delete posts. We also instructed participants to add a “Q” mark at the beginning of any question posts.
- 3rd floor: At the beginning of Period 5, all posts from Period 4 were deleted. We then implemented the same conditions for the 2nd-floor whiteboard, including periodic deletion and the instruction to add a “Q” mark to the beginning of questions. Additionally, we drew dividing lines on the whiteboard to separate different posts clearly (see Figure 3).

B. Observation Results

1) Whiteboard Usage

To investigate whether spontaneous posts would occur on a blank whiteboard without any specific instructions, we counted the number of posts on each floor (2nd and 3rd) during a period of 1 day. Table 1 shows the results. We did not distinguish between types of posts, such as questions and replies.

Table 1 shows that there were no posts on either the 2nd or 3rd floor on Day 1. However, from Day 2 onwards, there were posts on both floors every day. This confirms that posting can occur spontaneously on a blank whiteboard without specific instructions and that replies and comments to posts can also occur naturally.

Table 1 also shows that the number of posts on the 2nd floor on Day 3 was extremely high. To clarify this, we investigated the contents of the posts on Day 3. As a result, we found that posts were looking for part-time jobs and soliciting club membership, and there were many posts on these topics. We believe this is because the observation period coincided with the week when the new university semester began. Furthermore, we analyzed the contents of the posts after Day 3 to investigate why the number of posts increased. As a result, we found that many of the posts were about club recruitment. We can infer that this is because several people saw someone recruiting for a club on the whiteboard on Day 3 and started imitating them. Furthermore, an analysis of the correspondence between each post, i.e., the structure of the topic, revealed that multiple posts responded to a single question. This shows that whiteboards without specific instructions are environments with a high degree of freedom regarding the content of posts, as multiple users can freely respond to a single question.

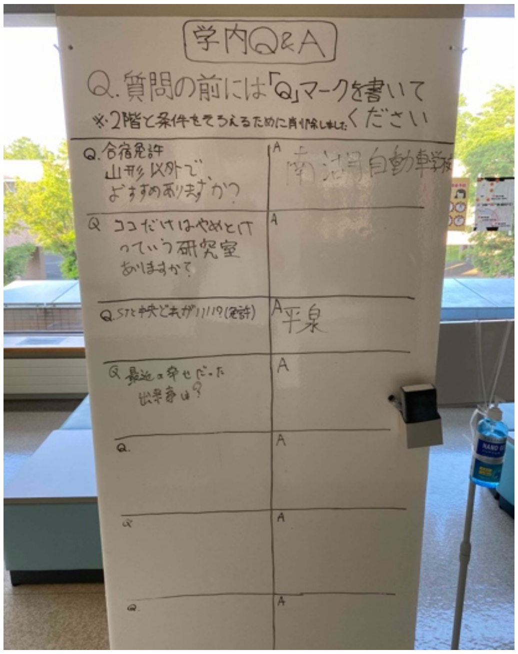

学内Q&A
Q.質問の前には「Q」マークを書いてください
※2階と条件をそろえるために削除しました
Q.合宿免許 山形以外で おすすめありますか？
A
Q ココだけはやめとけっていう研究室ありますか？
A
A 平泉
Q 最近の幸せだった出来事は？
A
Q.
A
Q
A
Q
A

Figure 3. 3rd-floor whiteboard in period 5

2) Whether Spontaneous Deletion Occurs and Effect of Regular Deletion

During Periods 2 (2nd week, Days 8-12) and 3 (3rd week, Days 15-19), we periodically deleted the whiteboards on the 2nd floor, while we did not delete the whiteboards on the 3rd floor. The periodic deletion was performed when a post was 2 days old, or the whiteboard was 80% filled.

First, we checked whether spontaneous deletion was done by users. On the 3rd-floor whiteboard, where no regular deletion was performed, the whiteboard was filled with posts by Day 11, and there was no space for new posts. However, users did not delete any of their posts, as shown in Table 2; there were no posts on the 3rd-floor whiteboard on Day 12. This suggests that users are unlikely to spontaneously delete their own posts even when they cannot post new ones without deleting something.

On the other hand, the 2nd floor whiteboard, where regular deletion was performed, had posts every day during the observation period, as shown in Table 2. However, the number of posts on Day 11 was only 2. This is likely because the whiteboard was 80% filled, and posts had less space than usual. These results suggest that regular deletion is effective in encouraging user posting.

3) Effects of Keeping Posts

At the beginning of Periods 3 and 5, all posts on the 3rd-floor whiteboard were deleted. As shown in Tables 3 and 5, the number of posts decreased for a few days after the deletion in both cases. However, once a new post was made, the number of posts tended to increase. On the other hand, on the 2nd-floor whiteboard, where we performed regular deletion, there was no day without posts except when there was little space left on the whiteboard. These results suggest that keeping a certain number of posts on a whiteboard can lower the barrier to posting.

4) Effects of Indicating Post Types and Gridlines

In Period 4, we added an explanation to the top of the whiteboard on the 2nd floor asking users to start question posts with the "Q" mark to clarify the types of posts. We did the same for the second and third floors in Period 5.

To investigate whether this increased question posts, we compared the number of question posts on the 2nd floor in Periods 1, 2, and 3 with the number of question posts in Periods 4 and 5 (see Table 6). The results showed that the number remained almost unchanged. This suggests that explicitly indicating questions with a "Q" mark and clarifying the types of posts did not have a significant effect.

In Period 5, on Day 57, we deleted all posts on the 3rd-floor whiteboard and drew a 2 × 7 grid to separate each post. We also added an explanation to the top of the whiteboard, as in Period 4 on the 2nd floor, asking users to start question posts with the "Q" mark. Additionally, we added this mark to each grid to indicate that each grid was an area for posting on a single topic. On Day 58, we observed that users had spontaneously changed the "Q"s in the right column of the grid to "A"s to indicate answers (see Figure 3).

TABLE I. NUMBER OF POSTS (PERIOD 1)

	Day 1	**Day 2**	**Day 3**	**Day 4**	**Day 5**
2nd floor	0	8	18	6	13
3rd floor	0	4	10	2	9

TABLE 2. NUMBER OF POSTS (PERIOD 2)

	Day 8	**Day 9**	**Day 10**	**Day 11**	**Day 12**
2nd floor	9	6	5	2	10
3rd floor	1	3	4	6	0

TABLE 3. NUMBER OF POSTS (PERIOD 3)

	Day 15	**Day 16**	**Day 17**	**Day 18**	**Day 19**
2nd floor	4	12	5	11	6
3rd floor	4	0	1	8	6

TABLE 4. NUMBER OF POSTS (PERIOD 4)

	Day 36	**Day 37**	**Day 38**	**Day 39**	**Day 40**
2nd floor	10	2	8	9	2
3rd floor	4	4	0	2	0

TABLE 5. NUMBER OF POSTS (PERIOD 5)

	Day 57	**Day 58**	**Day 59**	**Day 60**	**Day 61**
2nd floor	0	3	11	7	3
3rd floor	0	0	6	16	5

To investigate the effect of drawing gridlines, we compared the number of questions and answers on the third-floor whiteboard in Period 4 (with instructions to start question posts with "Q") with the number of questions and answers in Period 5 (with instructions to start question posts with "Q" and with gridlines drawn) (see Table 7). The result showed that the number of questions and answers increased in Period 5. This suggests that drawing gridlines was effective in promoting question-and-answer posts.

Another interesting observation is that there were no posts for a few days after the deletion on Day 57. However, on Day 59, posts started to appear again. On Day 60, there were 16 posts, of which 11 were replies to the "Q" marks (see Table 8). This was the highest number of posts per day during the observation period. We believe this is because the clear structure of the question-and-answer pairs, topic flow, and the pre-allocated writing space effectively promoted question-and-answer posts.

5) Continuation of Thread

Regular deletion was implemented on the 2nd-floor bulletin board during Period 2. When there was a reply to a new post, the original post was also kept for two days from the date of the reply. As a result, deletion was only performed on weekdays, so there were some threads where the deletion period was longer than usual due to weekends. Table 9 shows the transition of a thread. Since this thread was originally active, replies continued even after a week. To keep such active threads, it is considered necessary to change the deletion period depending on the reply status of each thread instead of deleting all threads uniformly according to a regular deletion rule.

6) Writing and Reading by Two or More People

We frequently observed several people browsing the whiteboard during the observation period, as shown in Figure 5. In some cases, we observed multiple people filling in the whiteboard (that is, only one person was actually filling in the whiteboard while multiple people were discussing the topic).

This situation is not seen in types of online communication, such as chat rooms or electronic bulletin boards. We believe it is important to take advantage of the benefits of analog communication, as in this case, to activate information sharing.

TABLE 6. NUMBER OF Q&A POSTS (2ND FLOOR)

	Period 1	**Period 2**	**Period 3**	**Period 4**	**Period 5**
Question	3	7	9	11	9
Answer	18	17	13	16	12

TABLE 7. NUMBER OF Q&A POSTS (3RD FLOOR)

	Period 1	**Period 2**	**Period 3**	**Period 4**	**Period 5**
Question	2	3	4	4	9
Answer	21	8	8	5	16

TABLE 8. NUMBER OF POSTS BY TYPE (PERIOD 5)

	Day 57	**Day 58**	**Day 59**	**Day 60**	**Day 61**
Question	0	0	4	3	2
Answer	0	0	2	11	3
Others	0	0	0	2	0
Total	0	0	6	16	5

This situation is not seen in types of online communication, such as chat rooms or electronic bulletin boards. We believe it is important to take advantage of the benefits of analog communication, as in this case, to activate information sharing.

C. Results of Questionnaire Responses

After the observation period, we conducted an online questionnaire survey open to all students at our university. The purpose was to gather information that could not be obtained from the observation, such as the attributes of the whiteboard users and their motives for writing on the whiteboards, as well as to gather information on features that should be incorporated into our electronic bulletin board. There were 208 responses. This number of responses represents approximately 10% of the total students.

TABLE 9. EXAMPLE OF THREAD STRUCTURE

	Content	**Post Date**
Post	I'm looking for a part-time job	Day 2
Replies	MITAKE (place name) recommended!	Day 4
	How about KAMIDOU (place name) Shabuyou (store name)?	Day 5
	Morioka is a good hotel!	Day 5
	How about purchasing?	Day 5
	Iwate Prefectural University Co-op HP→Part-time job information→ List of part-time jobs	Day 5
	Come to ODORI (place name) Happi (store name)!	Day 8
	Cake shop!	Day 8
	900 yen per hour to set out goods at Aeon (store name)!	Day 8
	Buckwheat noodle shop!	Day 12
	Home center	Day 12
	McDonald's is waiting!	Day 12

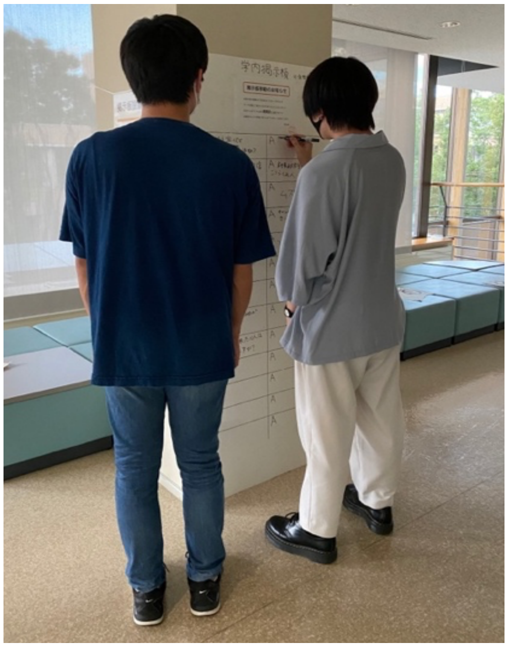


Figure 4. Multiple people writing on whiteboard

1) Dealing with Less Important Information

In response to the questionnaire question "Have you obtained necessary information or new findings from this whiteboard?", 58.2% of the respondents answered "No." The reasons were "I found it difficult to understand which information was important because there was too much unimportant information (44.8%)," "I felt the quality of the answers was low (22.9%)," and "I found it difficult to understand which information was important because there were many invitations to join club activities (9.5%)." Several responses said, "There was a lot of unimportant information, so I thought I could post any topic I wanted." During the whiteboard observation, we left unimportant information as is to facilitate posting. In consideration of these responses, we need to proactively address less critical information on our electronic bulletin board to achieve useful, high-quality information sharing.

2) Duration of Display of Posts

As already explained, during the observation period, we periodically deleted posts. Since we did not indicate the posting period on the board, we received a request in the survey to clarify when posts would be deleted. Considering this answer, the electronic bulletin board we design needs to tell users the remaining time before their posts are deleted.

3) Bulletin Board Location

When asked where the best place to place a bulletin board for information sharing is, 54.5% of respondents said the common building (near the entrance for students who walk or take public transportation), and 45.5% said in front of the shopping area. This was because "active responses can be expected," according to 71.7% of respondents. In other words, users believe that the bulletin board should be installed in a high-traffic area, and they believe that the bulletin board will be more active if it is installed in a visible location. However, installing the board in a high-traffic area that is easily visible to others contradicts the phenomenon known as social embarrassment [14], in which users are less likely to take action because they are concerned about being seen by others.

D. Design Principles of Electronic Bulletin Board Based on Whiteboard Observations

This section describes how the observations of the whiteboard are reflected in the design of the electronic bulletin board.

1) Mechanisms to Promote Posting

The results of the whiteboard observation showed that the number of posts was low when there was no space for posting or when there were no posts. It was also confirmed that a system for periodically deleting posts while keeping some of the content effectively solved this problem. Therefore, the electronic bulletin board to be designed will incorporate a function for the system to delete posts periodically.

In addition, the questionnaire results showed that there was a request to clarify the posting period. We believe that directly showing the remaining time until deletion for all posts may confuse users regarding readability. To avoid confusion, we need to consider a way to indicate the remaining time until deletion indirectly. Therefore, we will incorporate a function that indirectly conveys the remaining time until deletion, or the elapsed time of the post, by changing the font color to different shades.

Furthermore, the questionnaire results showed that users want the bulletin board installed in a place with high traffic and visibility. However, we believe that some users may feel uncomfortable posting in a place where they are visible and that we need to consider social embarrassment. In this regard, we will install the electronic bulletin board in places that are visible and considerate of social embarrassment, as well as places where users can post without being seen, and we will confirm the usage status at each installation location.

2) Mechanisms for Improving Quality of Posts

To realize useful information sharing, it is necessary for users to post information they want to know about and for appropriate responses to be posted. In addition, for users to easily find the information they need from the bulletin board, the visibility and readability of questions and answers need to be maximized.

In this regard, the results of the whiteboard observation showed that it is important for the quality of information to have post spaces prepared in advance and to be able to distinguish between posts that are similar to questions and posts that are responses to existing posts. It was also found that this is effective in promoting posting. Therefore, in the electronic bulletin board to be designed, post space will be secured in advance as a grid, and questions and corresponding answers will be placed next to each other.

In addition, the questionnaire results showed that many users felt that there was too much unnecessary information and that it was difficult to find important information. In the whiteboard observation, we did not consider the importance of the post content, and we did not take any special measures other than periodic deletion. However, we will deal with low-importance information in the electronic bulletin board to be designed. As the importance of the content of a post varies from user to user, we will incorporate a function that allows users who have read the content to vote on whether it is important or not. Depending on the voting results, the time the post is displayed on the bulletin board can be increased or decreased according to its importance.

3) Mechanisms to Continue Threads

In the whiteboard observation, it was confirmed that some threads can last for more than a week. Although there is a possibility that threads will continue even with the automatic deletion function mentioned above, they will also be interrupted by the periodic deletion. Therefore, we will incorporate a process that reflects the response status of posts in the deletion period into the periodic automatic deletion function.

4) Reflecting Advantages of Analog Bulletin Boards in Electronic Bulletin Boards

During the observation, we observed elements such as multiple people reading the whiteboard while talking about the content of the posts and posting while talking. To reflect this advantage of analog bulletin boards in an electronic bulletin board, we will adopt a method for operating the board in which the content of the posts is displayed on a large display and posts can only be made from a terminal attached to the large display.

IV. DESIGN AND IMPLEMENTATION

In this section, we describe the electronic bulletin board we designed and implemented on the basis of our observations of whiteboard usage behavior described in the previous section.

A. Configuration and Usage Environment of Electronic Bulletin Board

This section describes an electronic bulletin board designed for lightweight and informal information sharing within a large organization on the basis of the whiteboard observation results. Figure 5 shows the configuration of the designed and implemented electronic bulletin board and the flow for processing posted data. Unlike a typical online bulletin board where new posts are added and displayed one after another, this bulletin board uses a method where new posts are displayed by overwriting the space where past posts have disappeared over time.

To incorporate the unique advantages of analog bulletin boards, such as the behavior observed in the observation where multiple people write while conversing about the content of the posts, the board we implemented is not a closed bulletin board online but provides a place for information sharing as a real bulletin board. Therefore, the system is designed to work only on the browser of a display terminal

(Raspberry Pi 4) connected to a large display. In addition, posts to the board can only be entered from a keyboard connected to the terminal. As described later, when posts, replies, and reactions are made, the results are stored in the Realtime Database, a database provided by Google's web development platform Firebase, via the Internet.

B. *Layout of Electronic Bulletin Board*

Our electronic bulletin board has a predetermined posting area arranged in a 3 × 3 grid. Specifically, as shown in Figure 6, it consists of two 3 × 3 grid-like post lists that are automatically switched every 50 seconds. Fortin et al. [8] explained that displays that appear to be frequently used while still having free space can attract people, so we decided to arrange the post list in a grid format instead of a list format. If all the grids on the post list are filled with posts and there is no space to post, users can manually switch to another post list using the switch buttons on the screen's left and right sides. After manually switching to another list, users can find an empty space and post.

Each grid is configured as shown in Figure 7, with the top half being for question-like posts and the bottom half for replies to the top posts. In addition, an input field and a post button are provided at the bottom of each grid. Furthermore, two reaction buttons, "I saw it" and "Delete it," are provided for each post in the top and bottom halves. The method of posting and the usage of the reaction buttons will be described later.

C. *Posting Questions*

To post question-like content on the board, select a grid with a post space on the post list in Figure 6 and click the input area at the bottom of the grid in Figure 7. Next, type the content you want to post using the keyboard connected to the large display terminal. After entering, press the Enter key or the Send button at the bottom right of the post area to post the content. The posted content will be displayed instantly.

As pointed out by Brignull et al. [15], to promote interaction with the system, it is necessary for the interface to be clear and visible from the beginning. Therefore, we decided to always display the input area and the Send button in the post area.

D. *Replies to Posts*

To reply to an existing post, enter and submit your response in the input area. The input process is similar to posting a question-like post.

If there is already one or more replies, a link, "Check the list (n other posts)" ("n" is the number of other replies), will be displayed above the input area (Figure 7). Click this link to the reply list (Figure 8) and view other replies. The reply list screen has a similar input area, allowing you to reply after checking other replies.

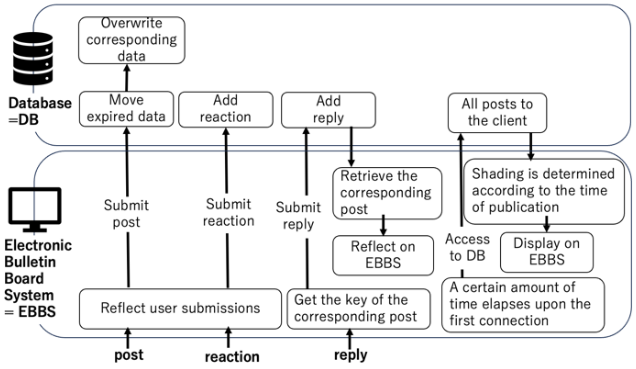
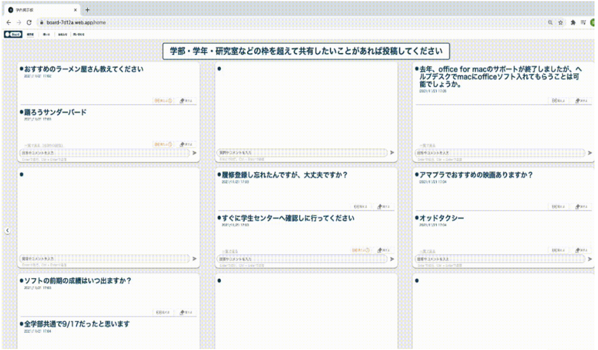

Overwrite corresponding data
Database =DB
Move expired data
Add reaction
Add reply
All posts to the client
Submit post
Submit reaction
Submit reply
Retrieve the corresponding post
Shading is determined according to the time of publication
Electronic Bulletin Board System = EBBS
Reflect on EBBS
Access to DB
Display on EBBS
Reflect user submissions
Get the key of the corresponding post
A certain amount of time elapses upon the first connection
post
reaction
reply

Figure 5. System configuration and data flow



学部・学年・研究室などの枠を超えて共有したいことがあれば投稿してください

- おすすめのラーメン屋さん教えてください
- 踊ろうサンダーバード
- 去年、office for macのサポートが終了しましたが、ヘルプデスクでmacにofficeソフト入れてもらうことは可能でしょうか。
- 履修登録し忘れたんですが、大丈夫ですか？
- すぐに学生センターへ確認しに行ってください
- アマプラでおすすめの映画ありますか？
- オッドタクシー
- ソフトの前期の成績はいつ出ますか？
- 全学部共通で9/17だったと思います

Figure 6. 3 × 3 grid-like post list

When there are multiple replies to a question-like post, replies posted less than 12 hours ago or with the most "I saw it" button presses will be displayed first. This is to keep replies that are attracting attention from many users always visible. It also prevents low-importance information, such as graffiti or harassment, from being displayed prominently.

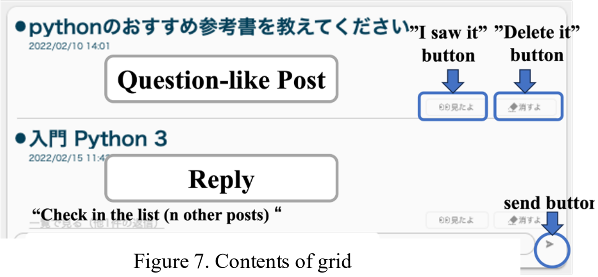
E. *Reaction Buttons*

Both question-like posts and replies each have two reaction buttons: "I saw it" and "Delete" (Figure 7).

- "I saw it" button: This button is used to indicate that the user has read a question that they found interesting, a reply that provides useful information, or simply to show that they have viewed the post.
- "Delete" button: This button is used to report inappropriate posts.

Initially, our bulletin board system used the commonly used UI of "good" and "bad" buttons. The "good" button was intended to be a way to show sympathy or to indicate that you had seen a reply, even if you did not reply yourself. The "bad" button was intended to be used to request that inappropriate posts or replies be deleted by a majority user vote.

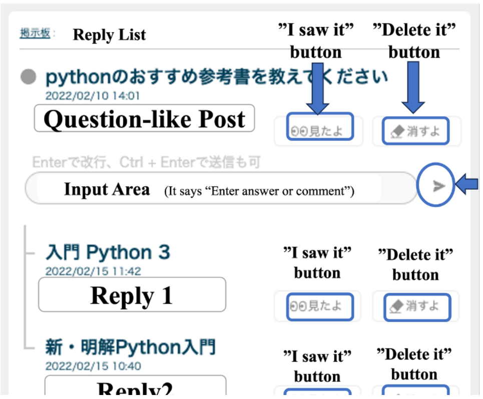
2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org



●pythonのおすすめ参考書を教えてください
2022/02/10 14:01

●入門 Python 3
2022/02/15 11:42

Figure 7. Contents of grid



掲示板 Reply List

● pythonのおすすめ参考書を教えてください
2022/02/10 14:01

Enterで改行、Ctrl + Enterで送信も可

入門 Python 3
2022/02/15 11:42

新・明解Python入門
2022/02/15 10:40

Figure 8. Reply list

However, we found that the names of these buttons did not clearly convey their intended purpose. Therefore, we changed the names and icons to "I saw it" and "Delete" in the middle of the operation period.

F. Posting and Reply Duration

All new posts and replies have a posting period. We initially considered a posting period of a few hours, similar to a station message board. However, after observing the whiteboards and the actual usage of electronic bulletin boards, we decided on a final posting period of 4 days. Additionally, posts made on Wednesdays and Thursdays were likely to expire on Sundays or Monday mornings when there are fewer people around. Therefore, we extended the posting period to 5 days for these posts.

Furthermore, the posting period is extended by 12 hours if the "I saw it" button is pressed and by 24 hours if a reply is received, and the posting period of a new post with the "Delete" button pressed is reduced by 24 hours. If the number of "Delete" button presses exceeds the threshold of 2, the post will be immediately hidden from the bulletin board.

For replies, if the number of "Delete" buttons pressed for each reply in the topic list exceeds the threshold of 2, that reply will no longer be displayed. The post will no longer be displayed, and the empty space will be available for other users to post again.

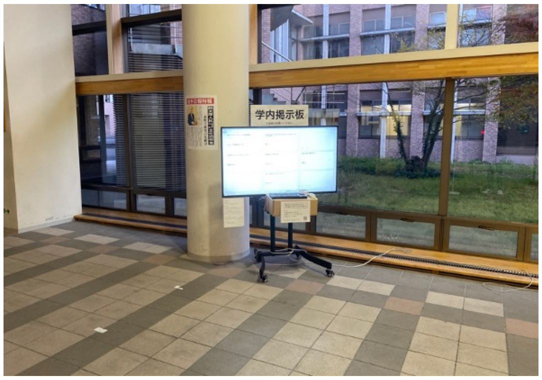


Figure 9. Our electronic bulletin board set-up

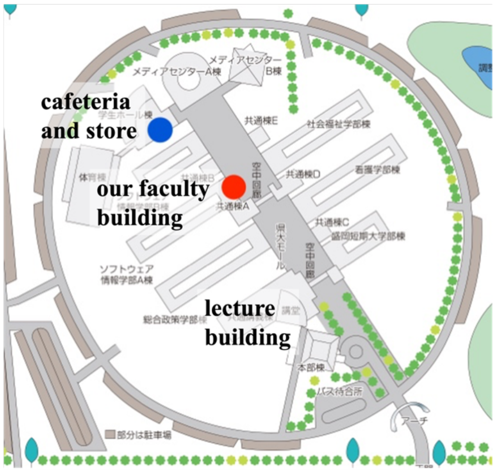


Figure 10. Campus plan of our university and locations of our electronic bulletin board

G. Changes in Font Weight and Background Color

In response to feedback from the survey conducted after observing the whiteboards, we implemented a system in which the font color fades in steps according to the remaining posting period (see Table 10). This was done to address the concern that users were unsure of when posts would be deleted.

Furthermore, we observed that users might not be aware of the fading font color. Therefore, we improved the system by changing the background color of each grid cell as well.

V. USER STUDY

In this section, we describe the procedures and results of a user study as an operational experiment of the electronic bulletin board we designed and implemented.

A. User Study Overview

We conducted a user study of the electronic bulletin board we designed and implemented for approximately 5 months to confirm its usage. At the start of this user study, we installed the board (Figure 9) in the area with the cafeteria and shop of the university where the authors belong (blue area in Figure 10). One month after the start, we moved it to the lounge space on the 2nd floor of our faculty building (red area in Figure 10).

At the top of the list of posts displayed on the board, we posted the following three explanations about the use of the board to encourage posting and the use of reaction buttons:

- "Please post anything you want to share beyond the boundaries of your department, year, or lab."
- "Posts will expire after a certain period of time."
- "Use the 'I saw it' button to indicate that you've read and are interested in a post, and the 'Delete it' button to indicate that a post seems unimportant."

To allow anonymous posting, the post entry field did not include a field for entering your name or affiliation.

In this study, we collected data on the content of the electronic bulletin board posts, the date and time of the posts, and reaction button logs (which buttons were used in response to which posts).

After the user study period ended, we conducted an online survey open to all students at our university to confirm the posting frequency and browsing status, which could not be confirmed from daily observation and the acquired data. The number of responses was 307. This number corresponds to approximately 15% of all students.

B. User Study Results

1) Posting Status

Figure 11 shows the number of new posts (question-like posts) and replies per week during the study period. Data from the two-week winter holiday (Week 14 and 15) is excluded. The total number of new posts during the period was 171, and the total number of replies was 351. The total number of new posts and replies ranged from 19 per week to 51 per week. Since there were almost no posts on Saturdays and Sundays, considering that there are 5 days in a week, there were about 3 posts per day in the week with the fewest posts and about 10 posts per day in the week with the most posts. This shows that our electronic bulletin board was used daily during the study period.

TABLE 10. CORRESPONDENCE BETWEEN REMAINING TIME FOR POSTING AND DEGREE OF SHADING

Remaining Period	**Shading Degree**
More than 3 days	1.0
Less than 3 days	0.7
Less than 2 days	0.5
Less than 36 hours	0.4
Less than 24 hours	0.3
Less than 12 hours	0.2

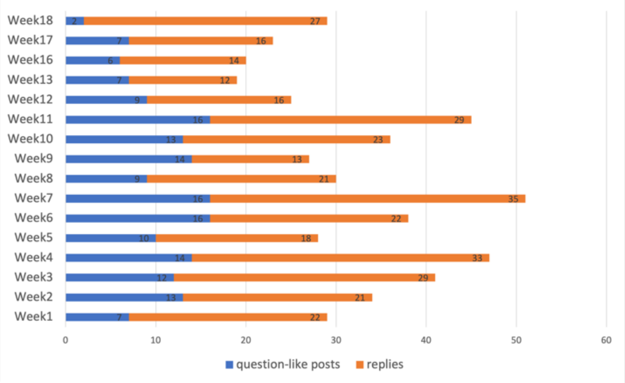


Figure 11. Number of new posts and replies per week

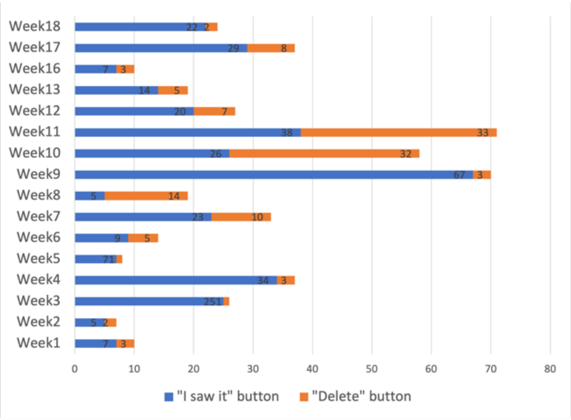


Figure 12. Number of times reaction button was used

C. User Study Results

1) Posting Status

Figure 11 shows the number of new posts (question-like posts) and replies per week during the study period. Data from the two-week winter holiday (Week 14 and 15) is excluded. The total number of new posts during the period was 171, and the total number of replies was 351. The total number of new posts and replies ranged from 19 per week to 51 per week. Since there were almost no posts on Saturdays and Sundays, considering that there are 5 days in a week, there were about 3 posts per day in the week with the fewest posts and about 10 posts per day in the week with the most posts. This shows that our electronic bulletin board was used daily during the study period.

We also found that the number of replies was overwhelmingly higher than the number of question-like posts every week. This suggests that multiple replies were made to each question.

In addition, Table 11 shows the frequency of use of the bulletin board as collected by the questionnaire. At the beginning of the installation, we were worried that some

2024, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

specific users might dominate the posting and replying, but we confirmed that the posting and replying were not biased towards specific users.

2) Viewing Status

Table 12 shows the results of the survey on the viewing status of the electronic bulletin board. During the user study period, 26 out of 307 respondents (8.5%) did not view the bulletin board at all, but 281 respondents (91.5%) had the experience of viewing it.

3) Reaction Button Usage

Figure 12 shows the number of times the "I saw it" and "Delete it" buttons were used per week during the study period. Data from the two-week holiday period at the end of the year (weeks 14 and 15) was excluded. The total number of times the "I saw it" button was used during the period was 338, and the total number of times the "Delete it" button was used was 132. The combined number of "I saw it" and "Delete it" button uses ranged from 7 to 71 per week. Since there were almost no posts on Saturdays and Sundays, this translates to an average of 1 use per day on the low end and 10 uses per day on the high end, assuming a 5-day work week. This suggests that the reaction buttons were also used on a daily basis throughout the study period.

Regarding which button was used more, the "Delete" button was temporarily used more only in weeks 8, 10, and 11. Otherwise, the "I saw it" button was used overwhelmingly more often. Looking at the usage ratio of each button for all 522 new posts and replies, the "I saw it" button was used for 31.4% of new posts and replies. The "Delete it" button was used for 9.4% of new posts and replies.

Furthermore, looking at the number of times each button was pressed, the "I saw it" button was pressed 23 times, and the "Delete it" button was pressed 6 times. Looking at the post content for which the "I saw it" button was pressed 23 times, it was for the response "We are currently developing an on-campus-only matching service. We would like to release it by the time we graduate, so please try it out" to the post "Where can I find a boyfriend/girlfriend?" There were two posts for which the "Delete it" button was pressed 6 times. One was a new post that simply said "a," and the other was for a new post that was posted multiple times with the same content. From this, it can be inferred that the "I saw it" button was used for posts that were considered useful, and the "Delete it" button was used for posts that were considered unimportant. This suggests that the buttons were used as we intended.

4) Structure and Duration of Threads

Table 13 shows an example of the thread structure formed by new posts and replies on our electronic bulletin board. As shown in this example, most new posts have multiple replies, indicating a one-to-many relationship between new posts and replies. The new post with the most replies received 21 replies. The duration of this thread (from the new post to the last reply) was 18 days. We also analyzed the duration of five other threads with more than 10 replies to new posts, and we found that all of them lasted for more than a week. These results suggest that threads on our electronic bulletin board tend to develop over a relatively long period of time, rather than progressing rapidly.

TABLE 11. POSTING AND REPLY FREQUENCY

	Number of People who Submitted	Number of People who Replied
Everyday	0	0
Several times a week	1	0
Once a month	0	4
Several times a month	2	3
Once every few months	7	9
Only once	20	17

TABLE 12. SURVEY RESULTS FOR ELECTRONIC BULLETIN BOARD VIEWING STATUS (N=307)

	Number of Answers
I did not view it at all.	18
I did not know there was a bulletin board.	8
I glanced at it when I passed by (not stopping).	116
I stopped in front of the bulletin board and viewed it.	89
I operated the mouse on the bulletin board and viewed it.	76

TABLE 13. EXAMPLE OF THREAD STRUCTURE ON OUR ELECTRONIC BULLETIN BOARD

	Content	Posting/Reply Day
Post	Tell me about a good niboshi ramen shop (other than Menya Iori and Niboshi SHINCHAN).	DAY 103
Replies	I don't know if it's only niboshi, but Hokuryu is delicious.	DAY 107
	Ikken!!!!!	DAY 107
	Nibo Shin.	DAY 107
	Hinotori	DAY 107
	Garitto	DAY 109
	Tairen no Ryu	DAY 109
	Samurai Boogie	DAY 109
	Birdmen one choice	DAY 113

5) Impact of Installation Location

To investigate changes in usage due to the presence or absence of foot traffic, we moved the bulletin board from the area with our university cafeteria and shop to the lounge space on the 2nd floor of our faculty building after the 6th week. The area with our university cafeteria and shop has a constant number of people passing through from around 11:00 to 18:00. On the other hand, the lounge space on the 2nd floor of our faculty building is only used to pass through when moving around the campus, and few people pass through during lecture hours.

Looking at the trend of the number of weekly posts and replies for the entire period shown in Figure 11, there was no significant change compared with before the move.

6) Use of Electronic Bulletin Board by Multiple People

During our whiteboard observations, we noticed that people often gathered around it to read posts, discuss their contents, and even create new posts together. We wanted to encourage similar interactions on our electronic bulletin board, so we decided to display posts on a large display and allow users to post only from connected devices.

To confirm whether people would actually use our electronic bulletin board in groups, we conducted two observation sessions at the bulletin board location during peak posting times (around 4 pm) during the user study period. We were pleased to observe that people did indeed use the bulletin board in groups on multiple occasions.

We interviewed the users using the bulletin board in groups to understand their motivations. They told us that they felt more comfortable using the board in a group because it allowed them to ignore the other people around them. Some even said they would not have used the board if they had been alone. These observations and interviews suggest that people may use electronic bulletin boards in groups to avoid social embarrassment.

VI. DISCUSSION

In this section, we further discuss the results of the user study of the electronic bulletin board described in the previous session.

A. *Posting Status*

The user study showed that most threads consisted of a single new post with multiple replies. There were no cases where one thread developed into another. However, an analysis of the collected data revealed some posts that appeared to be replies to replies. The current bulletin board system only allows replies to new posts, not replies to replies. All replies are displayed in the same thread. We may need to implement thread structure processing to address this limitation, such as moving replies to replies to separate threads. This would allow for more natural topic development and potentially facilitate creative information sharing, such as the generation of new ideas.

B. *Posting Method*

To recreate the advantage of analog bulletin boards where people can converse while posting, our electronic bulletin board can only be viewed on a large display, and posts can only be made from a terminal attached to the display. Observations of the usage of the electronic bulletin board confirmed our hypothesis that people would use it in groups, demonstrating that we successfully implemented an advantage of analog bulletin boards in the electronic version. However, the survey revealed a group of users who browsed the bulletin board using a mouse but did not post or reply. We believe this is due to the phenomenon of social embarrassment.

This is supported by free-response comments in the survey, such as “I find it difficult to stand in front of the bulletin board and read or write slowly (I’m self-conscious about other people’s eyes), so I would like to be able to access it from my smartphone or computer” and “I was self-conscious about people looking at me when I stood there and wrote. It might be easier to use if I could write freely from my own device.” To balance the advantages of analog bulletin boards with avoiding social embarrassment, we believe we need to provide two environments: one that uses the current large display and one that uses personal devices.

C. *Quality of New Posts and Replies*

From the survey results of the whiteboard observations, we identified the need to address low-quality information. In our electronic bulletin board, we implemented two reaction buttons, “I saw it” and “Delete it,” to allow users to evaluate the importance of each post. The display time of each post is then adjusted on the basis of these evaluations.

An analysis of the usage of each reaction button confirmed that the “I saw it” button was used for content that was likely to be important, while the “Delete it” button was used for content that was likely to be unimportant. This indicates that the reaction buttons function as intended.

In addition to reaction buttons, the number of replies to a new post can also be used as a clue to its importance. We, therefore, analyzed new posts with no replies. Out of 364 new posts, 33 met this condition. Of these 33 posts, the “I saw it” button was used for 2 posts, and the “Delete it” button was used for 8 posts. An analysis of the posts with “Delete it” button presses revealed that they included multiple posts of the same content, posts that provoked others, game opponent recruitment posts, and other content that some users might find unnecessary or offensive.

On the other hand, new posts with no replies and no reaction button usage were found to be solicitations, requests, or invitations that were difficult to reply to or questions so specialized that few people could answer appropriately. These posts will likely have a different level of importance than those with “Delete it” button usage.

Combining the number of replies to a new post with the reaction buttons can potentially improve the accuracy of evaluating low-quality posts. All new posts are categorized together, regardless of whether they are questions or other types of inquiries. By increasing the number of new posts, such as posts that require an answer, posts that do not require an answer but do require some action from the user, etc., we believe that the number of useful posts can be increased.

D. *Information Sharing Beyond Daily Communities in Large Organizations*

This study aims to support information sharing beyond small, daily communities, such as research labs, departments, or faculties, in large organizations like universities and companies. We examined the free-response survey results collected after the user study period to investigate whether our electronic bulletin board facilitated information sharing and interaction beyond daily communities.

The results revealed multiple responses indicating that information sharing and interaction beyond the usual faculty or department boundaries were indeed achieved. Some examples of such responses include:

- “I find it interesting because it gives me the opportunity to connect with people I wouldn’t normally connect with.”
- “Even though it was anonymous, I felt like I could interact with people from other departments, and it was enjoyable.”
- “I like it because it’s fun to look at and makes me feel connected to other students.”
- “I thought it was an interesting and useful tool because it allowed me to get opinions from people my age and get helpful information and words of sympathy. I didn’t post anything, but I enjoyed just looking at it.”
- “I found it very interesting to watch. I thought it was wonderful that there is a place for students to connect with each other, especially during the COVID-19 pandemic when communication has decreased. I thought it was great that you could connect with people inside and outside the university.”

These findings demonstrate that our electronic bulletin board successfully fostered positive impressions among users and facilitated meaningful connections beyond their usual communities.

E. *Strengths and Weaknesses of Our Electronic Bulletin Board*

On the basis of the results of the user study, we summarize the advantages and disadvantages of our electronic bulletin board. First, it has mechanisms to promote posting and to improve the quality of posting. The latter point is particularly important as our electronic bulletin board is intended for informal information sharing, which carries a high risk of lowering the quality of posts, and thus needs to be properly addressed. Second, it allows for our electronic bulletin board to be used by multiple users, which is an advantage of an analog bulletin board. The first advantage is realized by automatically deleting posts periodically and automatically extending and shortening of the display period using reaction buttons. The second is that users can post and view information via a large display, rather than from their personal terminals.

However, the disadvantage of this electronic bulletin board is that it is currently not possible to view and post from a personal terminal. This is in conflict with the second advantage mentioned above. The results of the user study show that users may avoid using or posting for a long time because they are afraid of being seen by others. To avoid this problem, we believe it is necessary to provide two environments, one through the current large display and the other through personal terminals.

VII. CONCLUSION AND FUTURE WORK

In this study, we hypothesize that by introducing an analog information sharing environment, such as a station bulletin board, into a digital information sharing environment, it will be possible to share information casually and informally across small community boundaries within a large organization. Based on this hypothesis, we conducted observations of whiteboard usage for about 2 months to incorporate elements that enable casual and informal information sharing in the analog world into digital information sharing. Then, based on the results of the whiteboard observation, we designed and implemented an electronic bulletin board. We then conducted a user study for about 5 months to verify the effectiveness of the various functions of the implemented bulletin board and to test the hypothesis.

The results of the whiteboard observations revealed the need for mechanisms to promote posting, improve post quality, and deal with low-importance posts. We also observed a situation unique to the analog world, where users converse about the content of posts in front of a whiteboard and post while conversing. Based on these whiteboard observation results, we designed and implemented an electronic bulletin board that allows users to post and view content through a large display.

The results of the user study showed that there were an average of 3 to 10 posts per day, including new posts and replies. In terms of viewing, the survey found that about 90% of the 307 respondents (about 15% of all students) had viewed the bulletin board at some point, confirming that our electronic bulletin board is being used daily as an information sharing tool. In addition, the "I saw it" reaction button was used for about 30% of all new posts and replies (552 posts), and the "Delete it" button was used for about 10%. Analysis of the message content in conjunction with the reaction buttons showed that they contribute to maintaining the quality of posts and judging the importance of posts, confirming the effectiveness of the implemented functions. Furthermore, the survey found multiple responses indicating that information sharing across departmental boundaries was possible. This confirms that the hypothesis that by incorporating an analog information sharing environment, such as a station bulletin board, into a digital information sharing environment, it would be possible to achieve casual and informal information sharing across small communities within a large organization, is correct. Additionally, it was found that many users had a positive impression of our electronic bulletin board.

However, we found that users may avoid using or posting for a long time because they are concerned about being seen. To solve this problem, we will study a electronic bulletin board environment that combines the current environment using a large display with an environment using personal terminals such as smartphones and PCs.

REFERENCES

[1] K. Ogura and R. Hoshi, "Design of Information-Sharing Media Based on Observation of Reading and Writing Behavior on Message Boards within Large Organizations," The 16th International Conference on Advances in Computer-Human Interactions, ACHI2023, pp. 60-64, 2023.

[2] Slack Technology Inc. Inspiring stories, real Slack customers. [online]. Available from: https://slack.com/customer-stories 2023.04.06.

[3] J. Chutrtong and W. Chutrtong, “Science Students’ Acceptance to use LINE Application in Laboratory Subject,” International Journal of Information and Education Technology, Vol. 10, pp. 227-231, 2020.

[4] R. Wang, L. Qiu, J. Cranshaw, and A. X. Zhang, “Meeting Bridges: Designing Information Artifacts that Bridge from Synchronous Meetings to Asynchronous Collaboration,” Proceedings of the ACM on Human-Computer Interaction, Vol. 8, CSCW1, pp. 1-29, 2024.

[5] S. Ejima, M. Oka, and T. Ikegami, “Concept of Keystone Species in Web Systems: Identifying Small Yet Influential Online Bulletin Board Threads,” Proceedings of the 11th ACM Conference on Web Science, pp.81-85, 2019.

[6] K. Nishimoto and K. Matsuda, “Informal Communication Support Media For Encouraging Knowledge-Sharing And Creation In A Community,” International Journal of Information Technology & Decision Making (IJITDM), World Scientific Publishing Co. Pte. Ltd., Vol. 6, No. 3, pp. 411-426, 2007.

[7] D. Snowdon and A. Grasso, “Diffusing Information In Organizational Settings: Learning From Experience,” CHI '02: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 331-338, 2002.

[8] C. Fortin, C. Neustaedter, and K. Hennessy, "Posting for Community and Culture: Considerations for the Design of Interactive Digital Bulletin Boards," CHI '14: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 1425-1434, 2014.

[9] Y. Chiba and K. Nishimoto, "An Intrablog-Based Informal Communication Encouraging System that Seamlessly Links On-Line Communications to Off-Line Ones," IEICE TRANSACTIONS on Information and Systems, Vol. E90-D, No. 10, pp. 1501-1508, 2007.

[10] A. Ardichvili, V. Page, and T. Wentling, "Motivation and Barriers to Participation In Virtual Knowledge-Sharing Communities Of Practice," Journal of Knowledge Management, Vol. 7, No. 1, pp. 64-77, 2003.

[11] A. Joinson, "Social desirability, anonymity, and internet-based questionnaires," Behavior Research Methods, Instruments and Computers, Vol. 31, No. 3, pp. 433-438, 1999.

[12] A. Joinson, "Self-disclosure in computer-mediated communication: The role of self-awareness and visual anonymity," European Journal of Social Psychology, Vol. 31, No. 2, pp. 177-192, 2001.

[13] S. Greenberg and M. Rounding, “The notification collage: posting information to public and personal displays,” CHI '01: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, pp. 514-521, 2001.

[14] R. Edelmann, "Social Embarrassment: An Analysis of the Process," Journal of Social and Personal Relationships, Vol. 2, No. 2, pp. 195-213, 1985.

[15] H. Brignull and Y. Rogers, "Enticing People to Interact with Large Public Displays in Public Spaces," Human-Computer Interaction INTERACT '03: IFIP TC13 International Conference on Human-Computer Interaction, pp. 17-24, 2003.