THE *FLAC* METHOD: DATA-FACILITATED DISCOVERY OF BUSINESS PROCESS IMPROVEMENT OPTIONS

Completed Research Paper

Tobias Fehrer, University of Bayreuth, Branch Business & Information Systems Engineering of the Fraunhofer FIT, Bayreuth, Germany, tobias.fehrer@fit.fraunhofer.de

Laura Marcus, Augsburg Technical University of Applied Sciences, Branch Business & Information Systems Engineering of the Fraunhofer FIT, Augsburg, Germany, laura.marcus@fim-rc.de

Maximilian Röglinger, University of Bayreuth, Branch Business & Information Systems Engineering of the Fraunhofer FIT, Bayreuth, Germany, maximilian.roeglinger@fim-rc.de

Uladzimir Smalei, Technical University of Munich, Munich, Germany, uladzimir.smalei@tum.de

Felix Zetzsche, FIM Research Center for Information Management, Technical University of Munich, Munich, Germany, felix.zetzsche@tum.de

Abstract

Business process improvement (BPI) is crucial to every business, as inefficiencies jeopardise an organisation's success. Predominant methods for BPI build on static process models, which are often incomplete, outdated, and lack execution-related insights. Process mining bears the potential to add execution-related insights into the process. However, organisations often lack the methodological expertise to apply process mining systematically to find process improvement options. Automating parts of BPI thus holds the potential to assist users without BPI expertise and enables data-driven BPI at scale. We introduce the FLAC method, which guides users in transforming conceptual BPI patterns into specific rulesets. Once transformed, they can be repeatedly applied to event logs to generate options for process improvement. An instantiation of the FLAC method on several BPI patterns and evaluation of its subsequent application to an event log confirmed its applicability and high relevance to practice by significantly reducing the time-to-insight.

Keywords: business process improvement, business process redesign, redesign pattern, situational method engineering.

1 Introduction

Business process management (BPM) enables organisations to operate effectively and efficiently through the continuous discovery, execution, analysis, and redesign of business processes (Dumas et al., 2018). Business process improvement (BPI) is a key part of the BPM lifecycle, aiming to redesign processes for incremental improvements (Dumas et al., 2018; Reijers and Limam Mansar, 2005). However, practitioners face challenges in their BPI endeavours: Collaboration between BPM method specialists with BPI expertise and business representatives with process context expertise is required to identify actionable and value-creating BPI options (i.e., specific ideas for improving a business process) (Zellner, 2013). The scarcity of BPM experts and business experts limit BPI scalability across organisations. This gap is also noticed in current research, highlighting the versatility and complexity of process redesign and calling for tools and methods to address these scalability challenges (Beerepoot et al., 2023; Zuhaira and Ahmad, 2021).

Automating BPI has the potential to enable BPI at scale (Beerepoot et al., 2023). However, current method toolsets offer limited support for generating BPI options. One frequently explored approach involves creatively applying established process improvement best practices. Textual descriptions of these practices (i.e., BPI patterns), drawn from field experience, can help identify potential improvement options (Fehrer, 2023). They are deemed to provide a simple entry point into BPI since they are understandable, describe clear ideas, and foster creative thinking about possible BPI options (Limam Mansar and Reijers, 2007). Yet, since BPI patterns are often described abstractly, their fit to a process requires specific knowledge of the process behaviour. Initial research provides promising support through the (semi-)automatic application of BPI patterns to a static process view (i.e., process models) (Netjes et al., 2008). However, there is untapped potential in incorporating process execution data into BPI.

Process mining (PM) is a dominant leading technique in process science, enabling evidence-based insights derived from business process execution data. While primarily used for analysis purposes, PM also holds vast potential for BPI (vom Brocke et al., 2021b). Despite its growing popularity, there is limited research on applying PM techniques to generate BPI options leveraging BPI patterns (Fehrer et al., 2022). In their regular conceptual representation, BPI patterns are impracticable for automation and scaling. To bridge the gap, there is a need to transform BPI patterns into specific rules that incorporate process data to enable automatic application to a process. This research aims to explore the specifications of such rules that match BPI patterns in content while being programmable and automatically executable as part of a BPI project. We hence formulate our research question as follows:

> *How can BPI patterns be transformed into programmed rulesets that might facilitate the automated development of redesign options in a BPI project?*

We address this research question by proposing *FLAC*, a method for translating BPI patterns into programmable rulesets. For example, from an abstract BPI pattern 'Empower' that is defined as "give workers most of the decision-making authority and reduce middle management" (Reijers and Limam Mansar, 2005, p. 301), we can derive a specific programmable rule: 'select pairs of activities executed in a sequence where the latter is executed by a resource from a higher hierarchy level' (see Table 2 for details and other rules of the corresponding ruleset). Our method suggests a scheme that aims to identify *F*itness, *L*ocation, *A*ttribute, and *C*onstraint rules, which replicate the application of a BPI pattern to a business process. This approach changes the value proposition of conventional BPI patterns since it separates (1) extracting best practice knowledge and (2) applying this knowledge to a process to detect BPI options into two activities. This allows scalability and better resource allocation since the manual, BPI expertise-intensive first activity should be conducted only once. The second activity can then be automatically performed in multiple BPI projects, thereby releasing valuable resources of BPI experts. To our knowledge, no other work addresses this research question.

The remainder of the paper is structured as follows: Section 2 provides a background on BPM, BPI methods, and related artefacts. Section 3 outlines our research design following the design science research method (DSRM) and situational method engineering (SME). In Section 4, we first propose design objectives (DOs) that guide the development of the *FLAC* method, present the method itself, and show its instantiation within a BPI project. Section 5 describes several evaluation activities. We instantiated *FLAC* to translate BPI patterns into programmable rulesets, programmed them as a prototype, and demonstrated the prototype's applicability and feasibility in a sample process. Throughout the method development, we focused on understandability, usefulness, ease of use, and fidelity with real-world phenomena, which are essential for evaluating method artefacts (Sonnenberg and vom Brocke, 2012). These aspects were evaluated both quantitatively and qualitatively during expert workshops. Section 6 discusses the evaluation outcomes and concludes by summarising our contribution, insights, limitations, and avenues for further research.

2 Related Work

The BPM discipline seeks methods, techniques, and tools to facilitate process redesign. Process redesign can be approached from a radical, explorative and ground-shaking perspective or with a continuous, exploitative and incremental ambition (Dumas et al., 2018; Gross et al., 2019; Rosemann, 2014; vom

Brocke et al., 2021a). While it appears appealing to thoroughly reconsider business processes and search for fresh value propositions in the face of ongoing change (Rosemann, 2014), there is a pressing need to consistently refine existing processes and seek enhancements for what already exists while maintaining the overall goal (Zellner, 2013). Using the term BPI, this research focuses on the incremental and exploitative modes. Incremental BPI seeks to identify improvement options by evaluating the effects of changes on process behaviour and performance goals (Zellner, 2013). The creativity to find BPI options can be enhanced by adopting best practices observed and collected from successful BPI projects in the field as so-called BPI patterns (also referred to as redesign heuristics or redesign patterns). BPI patterns "suggest particular changes to an existing process to influence its operation in certain ways" (Limam Mansar and Reijers, 2007, p. 193). Examples of BPI patterns include empowering the workforce to make decisions independently instead of seeking supervisor approval or parallelising tasks that can be worked on independently. Both examples aim to reduce processing time, while BPI patterns may also address other performance objectives (e.g., cost and quality) (Dumas et al., 2018). Several researchers have observed and documented BPI patterns from the field, facilitating their use in academia and practice (Limam Mansar and Reijers, 2007; Fehrer, 2023).

Extensive research has investigated using and applying BPI patterns (Jansen-Vullers and Reijers, 2005; Netjes et al., 2010). Jansen-Vullers and Reijers (2005) propose a generic four-step procedure for their application: (1) Identification of parts of the process that benefit from specific BPI patterns, resulting in a list of relevant patterns and focus process fragments. (2) Decision which (combinations of) BPI pattern(s) applied to a process fragment (BPI options) are relevant to the overall performance goal. (3) Creation of scenarios for each BPI option and evaluation of their feasibility and impact (compared to the baseline scenario). (4) Selection of BPI options to be considered in improving the process. Enabling increasingly automated applications of BPI patterns bears the potential for scaling and democratising this knowledge-intensive four-step procedure (Beerepoot et al., 2023). Adopting this procedure, further approaches seek to guide and semi-automate the application of BPI patterns when working with modelled processes (Netjes et al., 2008; Fehrer et al., 2022).

Since the approaches above require (up-to-date) process and simulation models, they are limited in their applicability and usefulness. After PM adoption has primarily focused on process discovery and compliance checking use cases, research calls for methods that bridge the gap between insight and BPI action (Park and van der Aalst, 2022; vom Brocke et al., 2021b). Niedermann et al. (2010) enrich static process models with data from a customised process data warehouse system to suggest BPI patterns. Their approach is limited in scalability since it is tailored around a specific process that must be executed in that system. A more flexible view of processes is supported by PM techniques that abstract process execution data into a standardised event log format before analysis (Park and van der Aalst, 2022). Several authors utilise PM to analyse processes for occurrences of BPI patterns or workarounds (i.e., friendly deviance) (Cho et al., 2017; van der Waal et al., 2022). However, beyond mere analysis, PM should be utilised to turn insights into tangible improvements (Stein Dani et al., 2024). In our literature review, no approach has come to our attention that supports the facilitation of BPI patterns for generating BPI options directly within process execution data.

3 Research Approach

We adopt the design science research (DSR) paradigm to address our research question. DSR aims to generate prescriptive knowledge about the design of innovative artefacts (Gregor and Hevner, 2013; Hevner et al., 2004). Our artefact is a method that enables the transformation of the BPI patterns for their further use in the automated data-facilitated discovery of BPI options. We utilise DSRM (Peffers et al., 2007) as a general research method for building and evaluating the artefact and following six phases *(1–6)*. In adherence to the DSR evaluation framework by Sonnenberg and vom Brocke (2012), we undertake four evaluation activities (EVAL1–4) along the phases of DSRM. As for *(1) problem identification*, we identify and justify (EVAL1) the need for a method that enables BPI patterns for the automated data-facilitated discovery of the BPI options in Sections 1 and 2. We *(2) define DOs for a solution* in Section 4 through literature synthesis and insights gained from six 45-minute interviews with

domain experts. These experts, including researchers (doctoral candidates and post-docs in the BPM domain) and practitioners (IT architects, data analysts, and consultants), possess over three years of BPM experience, demonstrating familiarity with BPI patterns and PM. The defined DOs constitute the solution space for the artefact design, offering support for unaddressed problems (Peffers et al., 2007) and guiding the design and development phase. We utilise SME as the specific research method for *(3) design and development* of the *FLAC* method in Section 4. SME distinguishes two modes: method configuration and method composition (Bucher et al., 2007). Method configuration adapts a base method against the background of a specific development situation. In contrast, method composition selects and orchestrates artefact fragments concerning the situational needs for achieving a particular goal (Bucher et al., 2007). We aim to develop a method for translating BPI patterns into programmable rulesets that will enable discovery of BPI options based on event data. We assemble existing method fragments from the BPM discipline and the fields of BPI and PM and hence opt for the method composition mode. This mode requires three stages (Bucher et al., 2007). First, we identify situational characteristics and formulate DOs in Section 4.1. Second, we decompose generic artefacts into artefact fragments (we primarily rely on our literature review to get an overview of available artefact fragments). Third, we compose artefact fragments in a situational method. We successively describe method activities, techniques, and roles and compile the activities into a procedure in Sections 4.2 and 4.3 (Braun et al., 2005; Denner et al., 2018; Vanwersch et al., 2016). Feedback from the aforementioned expert interviews and comparative assessment of existing artefacts against the DOs were utilised to evaluate and iteratively improve the design specification (EVAL2). We instantiate *FLAC* in an artificial BPI project with a software prototype as a *(4) demonstration* to assess the feasibility of the design specification (EVAL3). The *(5) evaluation* phase is reported in Section 5, focusing on the EVAL activity EVAL4. In a series of workshops, a different set of BPM experts than those interviewed for the problem identification applied the software prototype to generate BPI options. This manuscript serves as the initial *(6) communication* of the research findings. We also provide materials related to the *FLAC* instantiation (i.e., programmable rulesets, programmed rulesets, and automatically generated data-facilitated BPI options).

4 Design Specification

4.1 Specification of method requirements and design objectives

The interviewed experts and the literature agree that BPI requires significant time and expertise. They emphasise the need for practical methods, where an initial effort of structuring, translating, and implementing available BPI expertise may be justified by the subsequent high level of transferability and reusability that drives democratisation for conducting BPI, which can help scale BPI. Hence, we derive:

(DO.1) Democratisation and acceleration of BPI. The desired artefact should accelerate BPI projects and reduce the required expertise to conduct them.

Best practices are considered helpful for BPI. Interview participants stressed the importance of establishing a proper translation method based on their experience with BPI patterns. They rejected the idea of a one-off translation effort for an extensive set of BPI patterns. This decision was motivated by the unlimited variety of BPI patterns and their application possibilities. Hence, a translation method is deemed more valuable. Considering this feedback, we formulate DO.2 as:

(DO.2) Facilitation and customisation of BPI patterns. A suitable artefact should facilitate the adoption of accumulated BPI knowledge preserved in BPI patterns and enable case-specific adaptation of high-level BPI ideas for specific processes.

The expert feedback confirmed the literature-based hypothesis that process data is a relevant and trustworthy source of process information for BPI. Thus, DO.3 demands:

(DO.3) Incorporation of process data. Both event logs and context data serve as sources of information for tapping into the improvement potential of business processes. An artefact should capitalise on the information available, encompassing details about processes such as historical execution data and their contextual background.

The project type and an invariant context are relevant factors in composing the situation for applying the method (Bucher et al., 2007). The BPM context framework describes four *context factors* from which we derive the method context adjusted to the DOs: The method should be geared towards exploitative BPI projects (*goal dimension*). Considering the *process*, we focus on core and support processes with medium variability. The process must be repetitive to leverage data science techniques, and its execution data must be captured in event logs. Regarding the *organisation*, we focus on medium or large organisations that actively conduct BPM and intend to scale BPI. The method is agnostic of the *environment*.

4.2 *FLAC*

Traditional use of BPI patterns follows the execution of several consecutive stages (see Section 2). These stages can be carried out either manually or automatically. Achieving complete automation of all stages represents the ideal state for automated BPI, though this is unrealised. Our objective is to automate a specific activity – generating BPI options.

For this, we split the traditional BPI procedure into two parts. First, the *FLAC* method guides the translation of BPI patterns into intermediate programmable and final programmed rulesets (see Figure 1). Second, the programmed rulesets are repetitively applied in BPI projects to automatically generate BPI options, as demonstrated in Section 4.3.

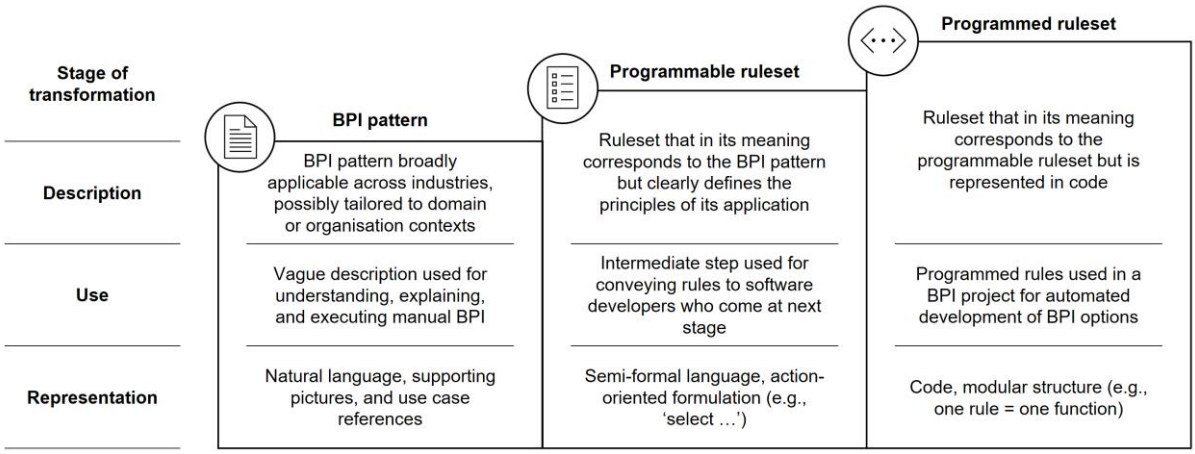

Figure 1. The stages of transformation of a BPI pattern using the FLAC method.

To initiate the method, users select a BPI pattern for the translation. In four steps, rules that focus on the Fitness, Location, Attribute, or Constraint perspective on the BPI pattern are determined (see Table 1). The structuring draws from established methodologies in BPI pattern application such as Jansen-Vullers and Reijers (2005), carefully tailored in line with method requirements and DOs. Each step of the method offers a unique perspective for investigation, thereby enriching the BPI pattern with greater clarity and substance. For each perspective, we describe (a) a purpose, defining the rationale for examining the BPI pattern from this perspective, (b) a guiding question that provokes the search for relevant rules, and (c) possible answers to facilitate the ideation of rules. These steps can be carried out individually or collaboratively using brainstorming or case studies. The goal of the activity is to produce a *programmable* ruleset.

Step 1: Derive Fitness rules (F): This perspective determines if applying a particular BPI pattern can positively impact the process concerning the performance objective or whether it is already sufficiently explored and thus not relevant to the process. Later, in a BPI project itself, the result of the check of the Fitness rule will guide the decision to either 'continue' or 'stop' further search for BPI options. Hence, it is set as the first rule to save computing power. Defined Fitness rules, like all other rules, can undergo testing against individually established thresholds. These thresholds are determined empirically, with suggested values outlined by Netjes et al. (2008). Ensuring that organisations define their criteria for these thresholds is essential, as it reflects their unique circumstances, objectives, and information.

Step 2: Derive Location rules (L): This perspective provides aid in identifying sets of instances (process and event log elements) relevant to the BPI pattern. At this point, all instances are considered, irrespective of their attributes, with the sole emphasis on their presence in the process. Depending on the BPI pattern being analysed, the locations are individual activities, sub-graphs, groups of process instances with shared commonalities (i.e., process variants), or resources. Each identified location gets a score between minimal 0 and maximal 1 to demonstrate the relevance of application of the BPI pattern at this specific location.

Step 3: Derive Attribute rules (A): This perspective functions as ranking and assessment support for the locations (entities) identified in the previous step. Each Attribute rule is quantified, assigning a relevance score between minimal 0 and maximal 1 to every location. The cumulative score across all Location and Attribute rules determines the relevance of applying the BPI pattern to a specific location, allowing for ranking all potential locations. To streamline the suggested BPI options, excluding locations that receive 0-score ratings for specific attributes can be prudent.

Step 4: Derive Constraint rules (C): This perspective further guides the user regarding the pre-ranked list of locations. Which constraints and possible limitations should be considered before/during applying the BPI pattern? Constraint rules function as a system of filters. Since our aim is not a fully automated BPI approach, and a user evaluates the BPI options, we retained locations with violated constraints in the final list. This decision stems from recognising that constraints may vary depending on the user's judgment. For instance, if the application of a BPI option requires a 10 % increase in one of the resources, the user can either adhere to the current resource constraint, deeming the BPI option unsuitable, or opt to lift the constraint, acquire additional resources, and proceed with implementing the BPI option.

The user transforms a BPI pattern into a programmable ruleset by progressing through the four perspectives, as displayed in Table 1. Following the attainment of the programmable ruleset, the subsequent activity involves programming the rules. This activity can be executed in any suitable software environment using tools available to the user, such as PM tools, process query languages, or libraries focused on process analysis. We recommend programming the rulesets in a modular structure, where each rule corresponds to one function. This is for the individualisation, exclusion, or addition of separate rules without impacting the functionality of the entire code. Upon completion, a BPI pattern is translated into a programmed ruleset and stored in the collection (see Figure 2), ready for application to a suitable process event log.

We have designed the method to be versatile for various user groups. On the one hand, users in academia can leverage the *FLAC* method to translate collections of BPI patterns into programmed rulesets. On the other hand, industry users may concentrate on applying the *FLAC* method to industry- or organisation-specific adaptations of BPI patterns. The latter group also has the flexibility to customise the programmed rulesets developed by academia to align with the organisation's demands and context and add them to the internal collection of rulesets (see Figure 2).

Method users must acknowledge that all rules, and consequently, the automatically generated data-facilitated BPI options, are influenced by the users' knowledge, experiences, and preferences during translation. Even though individuals with limited BPM and BPI expertise can create rules, the quality of the results depends on the knowledge and experience of the participants involved in the transformation project. However, including this one-time transformation in the BPI procedure allows the process to be separated into more and less BPI expertise-intense parts and the output of the most expertise-intense part to be reused (see Section 4.3 for details). As indicated in Table 1, we recommend involving Senior BPM experts and process owners in rule derivation and engaging software engineers in ruleset programming.

Activities *tasks of the method*	**Techniques** *detailed instructions on how to execute activities*	**Tools** *means supporting the execution of activities*	**Roles** *users executing activities*	**Outputs** *results of activities*
Derive *FLAC* rules	• Find specific rules that are hidden behind the vague description of the BPI pattern • Possible formats: brainstorming, brainwriting, case studies, etc. • Guidance and facilitation: Purpose, Guiding question, Possible answers	• BPI pattern catalogues with natural language descriptions of best practices • BPI case study reports	• Senior BPM expert (method expertise) • Process owner (subject matter expertise) • BPM researcher	**A programmable ruleset** that corresponds to the specific BPI pattern
1. Fitness perspective	**Purpose:** Understand whether applying this BPI pattern is relevant or has already been sufficiently explored. **Guiding question:** Which **process-level indicators** show if the BPI pattern is already sufficiently applied? **Possible answers:** process-level nominal values (e.g., number of gateways), process-level ratios (e.g., level of idle time).			
2. Location perspective	**Purpose:** Understand in which areas of the process it makes sense to apply this BPI pattern. **Guiding question: Where precisely** within the process can the BPI pattern be applied? **Possible answers:** specific activity (e.g., every activity that gets input from two others), sequence of activities (e.g., two consecutive gateways), process variant (e.g., process variants that include more than ten activities), resource.			
3. Attribute perspective	**Purpose:** Understand which attributes define whether applying the BPI pattern at a specific location is reasonable. **Guiding question:** Which **specific attributes** show how sensible the application of the BPI pattern at the specific location is? **Possible answers:** attributes of activities (e.g., duration of an activity), relations between activities (e.g., the same set of resources executes both activities), attributes of instances (e.g., the average number of resources involved in one instance).			
4. Constraint perspective	**Purpose:** Understand which constraints should be considered when applying this BPI pattern. **Guiding question:** Which **constraints and possible limitations** should be considered before/during applying the BPI pattern? **Possible answers:** DOs during redesign (e.g., DO consider required merging time), DON'Ts during redesign (e.g., DO NOT violate data dependencies).			
Program ruleset	Implement and test the ruleset using the modular structure (each rule as a function, method, etc.)	PM tools and libraries (e.g., PM4Py, Celonis PQL query language)	• Software engineer • Process owner	**A programmed ruleset** that corresponds to the specific BPI pattern

Table 1. The FLAC method for transformation of BPI patterns into programmed rulesets.

4.3 Enacting *FLAC* in BPI projects

The advantage of using *FLAC* method for BPI is that the traditional manual BPI procedure can now be divided into two parts: preliminary transformation of BPI patterns into programmed rulesets and the BPI project itself (i.e., derivation of BPI options for a specific combination of business process and BPI

pattern), as visualised in Figure 2. As a result of this split, the expertise-intense transformation of a BPI pattern into a programmable and programmed ruleset is manually conducted only once using *FLAC* method. Its output is stored in the collection of rulesets and might be reused in multiple BPI projects to automatically generate the initial set of BPI options and lower the barrier of BPI expertise needed for conducting the BPI project itself. Hence, the amount of time and effort required to translate BPI patterns into programmed rulesets is justified by the reusability of its output (DO.1). In a BPI project, one should apply this programmed ruleset against the event data to automatically retrieve data-facilitated BPI options. A complete BPI project itself can be split into five consecutive activities:

(1) Select business process: first, the project team selects a process that is the focus of the BPI project. Focusing on core and support processes with medium and high variability makes sense. The process should be highly repetitive to leverage data science techniques, and its instantiations should be captured in event logs. The method is agnostic of the process scope (i.e., intra- vs. inter-organisational processes) and industry (manufacturing, service, public sector, etc.) if process execution data is available.

(2) Select BPI pattern from the collection: second, the project team selects a BPI pattern from one of those available in the collection to apply to the event log. They use BPI pattern descriptions and estimated impact on the process performance. A BPI pattern was either already translated into a programmed ruleset that can be adapted to the selected process or should be translated (following the transformation process that is in detail described in Section 4.2).

(3) Customise programmed ruleset: after selecting the BPI pattern, corresponding programmed ruleset should be customised. BPI project settings bear varying data structures. Hence, going through the rules and adapting them to the data structures and context might be required from both conceptual (programmable ruleset) and implementation (programmed ruleset) perspectives.

(4) Run rule-checking algorithm: after customisation, the ruleset for the respective BPI pattern is applied to the event data to generate BPI options automatically. Following a defined order and using different rule types to reduce computing effort is advisable. This ensures efficient processing by first checking whether the BPI pattern fits the process before proceeding with searching, ranking, and filtering options (see Section 5.1).

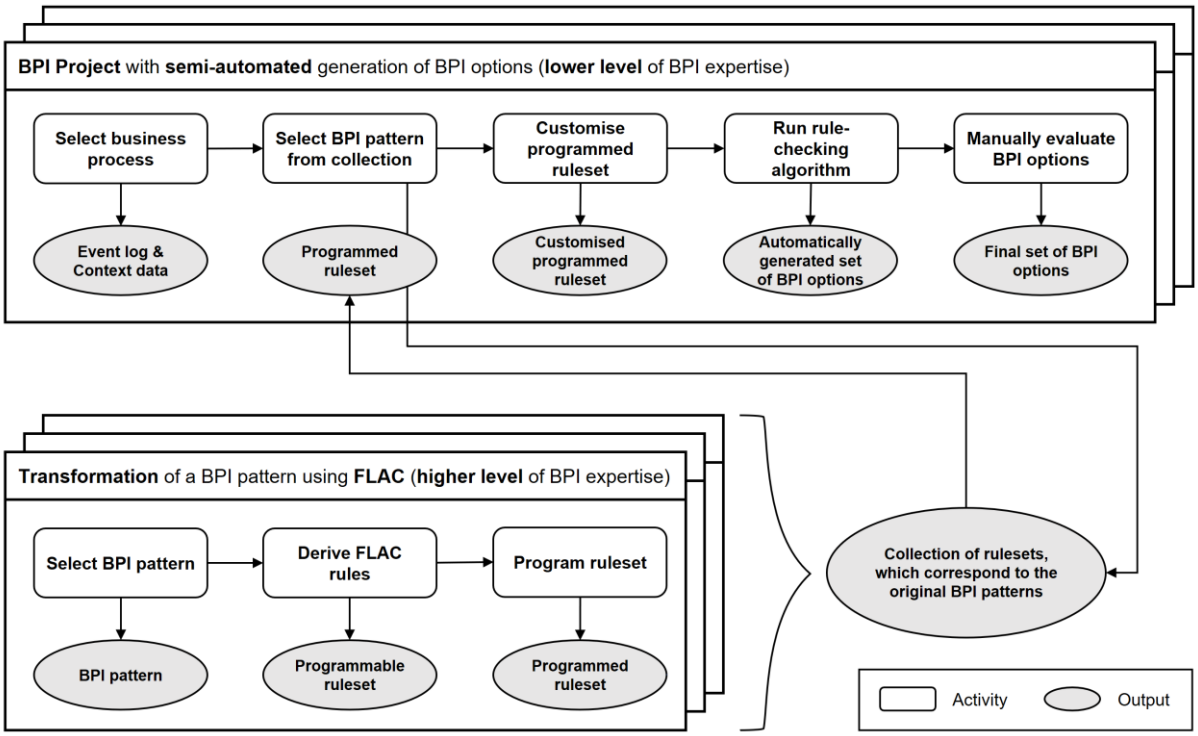

Figure 2. Schematic overview of the transformation projects and BPI projects.

(5) Manually evaluate BPI options: with the ranked list of BPI options, the project team discusses the possible process improvements. By going through all the BPI options and filtering out unsuitable or counteracting ones, the team ends up with a set of BPI options for implementation.

At the end of the BPI project, the team can start iteratively applying selected BPI options and tracking their impact as a part of continuous improvement. The whole procedure is repeated for other processes and BPI patterns. It can also be repeated for the same combination of process and BPI pattern after a certain time when the changes showed their effect.

5 Demonstration and Evaluation

5.1 Application of *FLAC*

We applied the *FLAC* method and instantiated an exemplary BPI project on multiple publicly available event logs[1] for development purposes and a real-life purchase-to-pay (P2P) process for demonstration purposes. We translated four BPI patterns (parallelism, case assignment, resource empowerment, and activity automation) into programmable rulesets and implemented them in Celonis EMS[2].

We selected BPI patterns that vary in their primary improvement objective, event data entities (activities, pairs of activities, process variants), and their level of abstraction. For each of the four selected BPI patterns, three authors independently applied the *FLAC* method following the purpose, guiding question and possible answers provided in Section 4.2 and transformed the pattern (based on literature descriptions and their BPI experience) into programmable rulesets. The consolidated rulesets are presented in Table 2.

To convert the programmable rulesets into programmed rulesets, we implemented them in the Celonis EMS Machine Learning Workbench, a computing platform based on Project Jupyter enabled for the work with process data. The source code for the rules is available online[3]. The code is modularly structured, where single rules are implemented as functions that are later brought together to check specific BPI patterns (this allows organisations to dynamically rewrite/exclude/reuse/extend rules based on the context). Some functions we programmed use only event logs as input, while others additionally accept contextual data (e.g., organisational hierarchy of process resources for the empowerment pattern). Depending on the *FLAC* perspective, functions generally take different forms:

- Functions addressing Fitness rules typically calculate a process-level ratio or nominal value, compare it to a threshold, and decide whether to continue checking other rules.
- Functions addressing Location rules define and rank-filter a dataset of entities (single activities, groups of activities, etc.). For example, for the parallelism pattern, we initialise an activity-activity matrix assigning 1 to the pairs that only occur in sequence, 0 to the ones that only happen in parallel, and a score between 0 and 1 to pairs with uncertain degrees of parallelisation.
- Functions addressing Attribute rules assign a normalised score between minimal 0 and maximal 1 to each entity identified in the Location functions.
- Functions addressing Constraint rules define whether implementation of the BPI option will result in the violation of ‘as-is’ constraints. The result of checking a Constraint rule might be True (‘as-is’ constraint will not be violated), False (‘as-is’ constraint will be violated) or ‘Null’ (it is not possible to decide based on provided data). ‘Null’ values additionally warn users at this point that additional attention is needed. We opt to keep all entities in the final list of BPI options because the user might be willing to change ‘as-is’ constraints and implement some of the BPI options beyond the current constraints level.

[1] Task Force on Process Mining event log collection: https://www.tf-pm.org/resources/logs

[2] Celonis EMS Machine Learning Workbench: https://docs.celonis.com/en/machine-learning-workbench.html

[3] Link to *FLAC* instantiation source code: https://doi.org/10.6084/m9.figshare.24560257.v1

Ranking of the BPI options in the algorithm is based on the total score, which is the sum of the scores from all the checked Location and Attribute rules. For this, the output of checking these rules should be a normalised score (e.g., between 0 and 1), where the lowest [highest] score always corresponds to the least [most] favourable condition for the application of the BPI pattern.

In the prototype implementation, all ruleset functions are combined into a module to assess a BPI pattern's applicability.

BPI Pattern	**Improvement Objective***	**Programmable Rules**
Parallelism (Reijers and Limam Mansar, 2005; Netjes et al., 2008)	+ Time	[FITNESS 1] Check that the process is not already overloaded with parallel activities, i.e., current level of parallelism < 0.4.
		[LOCATION 1] Select pairs of activities executed in sequence without overlap. Activities already executed in parallel should be excluded from option list.
		[ATTRIBUTE 1] Estimate the mean execution time of activities in pairs. Similarity in mean execution time is preferable.
		[CONSTRAINT 1] Check that data dependencies within the process are not violated. Activities do not depend on each other's output. [CONSTRAINT 2] Check that the time saved from the parallelisation of two activities is greater than the time required to merge the results of these two activities. [CONSTRAINT 3] Check that sufficient resources are available to execute activities in parallel (if the same resource executes both activities, extra resources might be needed).
Case assignment (van der Aalst and van Hee, 2002; Reijers and Limam Mansar, 2005)	+ Time + Quality	[FITNESS 1] Check that the process variant is not already executed using the 'case assignment' BPI pattern, i.e., more than one resource is involved in the execution of the process variant.
		[LOCATION 1] Select process variant overall.
		[ATTRIBUTE 1] Estimate the number of resources executing the process variant (normalised by the number of activities per process variant). Low values are preferable because they indicate that fewer resources are already assigned to case activities. [ATTRIBUTE 2] Estimate the number of transitions from one resource to another within the process variant (normalised by the number of activities per process variant). High values are preferable because they indicate higher inefficiencies in the 'ping-pong' of the case between resources.
		[CONSTRAINT 1] Check that the resource has sufficient knowledge to execute new activities that other resources previously conducted.
Resource empowerment (Buzacott, 1996; Netjes et al., 2008; Reijers and Limam Mansar, 2005)	+ Time + Cost + Flexibility	[FITNESS 1] Check that the process includes sufficient authorisation activities, i.e., current level of authorisation > 0.2.
		[LOCATION 1] Select activities that include authorisation signs in their names (e.g., '%authori%', '%approv%', etc.). [LOCATION 2] Select pairs of activities executed in sequence where the latter is executed by a resource from a higher hierarchy level (this might be a sign of authorisation).
		[ATTRIBUTE 1] Estimate the cost of a mistake for each activity. Low values are preferable.
		-
Activity automation (Netjes et al., 2008; Reijers and Limam Mansar, 2005; Nissen, 1996)	+ Time + Cost + Quality	[FITNESS 1] Check that the process is not fully automated, i.e., level of automation < 1.0.
		[LOCATION 1] Select activities with less than 100 % automation level. Activities with 100 % automation level should be excluded from option list.
		[ATTRIBUTE 1] Estimate the frequency of each activity. High values are better. [ATTRIBUTE 2] Estimate the mean execution time of each activity. Low values are better. [ATTRIBUTE 3] Estimate the number of resources executing each activity. High values are better.
		-

*) *Improvement objective* refers to the assumed impact on process performance described in the BPI pattern descriptions. (+) indicates a positive performance development (i.e., decreasing cost and time are positive developments). We refer to the performance dimensions described in the Devil's quadrangle of process performance in line with Reijers and Limam Mansar (2005).

Table 2. Instantiation of FLAC-derived rulesets for four BPI patterns.

5.2 Instantiation of a BPI project

Following Section 4.3, we instantiated our BPI project. First, we opted for a P2P process available within the Celonis Academic environment. This process is frequently targeted by PM initiatives and is commonly featured in educational and training materials. Hence, it serves as an ideal candidate for evaluation, given its widespread recognition and comprehensibility across diverse audiences. Despite its apparent standardisation, the P2P process presents ample opportunities for enhancement, owing to its abundance of cases.

Next, we selected a BPI pattern to test our event log against (for demonstration purposes, we use the example of parallelism, while details on the remaining three BPI patterns can be found online). In Section 5.1 (which logically corresponds to Section 4.2), we already conducted a one-time transformation of four BPI patterns into programmed rulesets. As 'Parallelism' was one of them, we only had to review the rules and adapt them to the P2P process. From a conceptual perspective, no changes were necessary to be considered for the programmable rulesets. The programmed rulesets were adapted to the attributes present in the event data. Following the order of the *FLAC* rules, the algorithm first checks the Fitness rule. For parallelism, the check involves verifying if the existing level of parallelism is below 0.4 (see rule [FITNESS 1] in Table 2). In the instantiation case for the chosen P2P process, it equals 0.29. Hence, the algorithm proceeds with the other rule types. As an outcome of reviewing other five rules (Table 3), the algorithm returns a list of four generated BPI options that are ranked based on their suitability to the considered process (for rule names, please refer to Table 2; for calculation details, please refer to the online appendix). In addition to presenting potential BPI options, the algorithm provides scores (for Location and Attribute rules) or values (for Constraint rules: 'True' if Constraint is not violated, 'False' if Constraint is violated, 'NULL' if there's no sufficient data to decide) for all assessed rules, substantiating that the output can be used in manual evaluation of BPI options and following discussions.

Rules for 'Parallelism' BPI pattern (Table 2) / **# BPI Option**	**[LOCATION 1] (score from 0 to 1)**	**[ATTRIBUTE 1] (score from 0 to 1)**	**[CONSTRAINT 1][a)]**	**[CONSTRAINT 2][a)]**	**[CONSTRAINT 3][a)]**	**Total Score**
1) Parallelise "Create Purchase Requisition Item" → "Create Purchase Order Item"	0.89	1.00	True	Null (time economy[b)] = 48 hrs; merging time[c)] unknown)	True	1.89
2) Parallelise "Change Currency" → "Record Goods Receipt"	0.89	0.96	True	Null (time economy[b)] = 144 hrs; merging time[c)] unknown)	True	1.85
3) Parallelise "Record Invoice Receipt" → "Clear Invoice"	0.96	0.35	True	Null (time economy[b)] = 250 hrs; merging time[c)] unknown)	False	1.31
4) Parallelise "Record Goods Receipt" → "Clear Invoice"	1.0	0.18	True	Null (time economy[b)] = 151 hrs; merging time[c)] unknown)	True	1.18

[a)] True = Constraint is not violated; False = Constraint is violated; Null = no sufficient data to decide.

[b)] Time economy is an estimate of the saved time because of executing activities in parallel instead of in sequence. The time of the shortest activity in pair is used as an estimate for time economy.

[c)] Merging time might be needed to bring together the output of two activities. No proxy for merging time was defined by the user for this process.

Table 3. Automatically generated set of BPI options for selected P2P process and 'Parallelism' BPI pattern (exemplary algorithm output).

Subsequent evaluation of BPI options relies on a combination of common sense and process knowledge. It is acknowledged that additional research may be necessary for specific rules that have not been automatically checked by the algorithm due to missing data (e.g., Null values in Constraint rules). We emphasise that the provided list should not be directly employed as an action plan. Instead, a project team should do a proper evaluation activity (including evaluation of business impact). However, the algorithm enables a significant reduction in time-to-insight by providing a list of BPI options ranked based on their suitability to the process.

The evaluation activity serves as a logical conclusion to the preparation of the BPI options list. Subsequently, the implementation phase begins, where selected options are applied to improve the process. As changes are made and new data is collected, the whole procedure can be iteratively repeated. This approach allows for ongoing refinement and consideration of other BPI patterns.

5.3 Evaluation of *FLAC* with experts

Following DSR principles, we conducted several demonstration and evaluation activities, ex-ante and ex-post. In the following, we detail ex-post evaluation in an artificial setting (Venable et al., 2012). We conducted five case study workshops involving seven BPM experts, four from academia (interviewed in pairs of two) and three from industry (interviewed separately). Our decision to work with this relatively small sample size was deliberate, as we aimed to include only experts with substantial experience with BPM, specifically BPI and PM. This helps ensure a high reliability of the evaluation results. The four academic experts are doctoral candidates in information systems (IS), with a particular research focus on BPM. Notably, three of the four have also been involved in consulting projects related to BPI. The three industry practitioners all possess backgrounds in IS and have accumulated several years of experience in BPM. Two of them work at large corporations and are responsible for PM initiatives within their organisations. This provides them with extensive experience in PM technology, process analysis, and BPI. The third practitioner is a software engineer at a medium-sized Software-as-a-Service organisation, currently responsible for adopting PM capabilities in their solutions. All the experts possess extensive knowledge in BPM and PM, partially with a strong emphasis on BPI. All experts also have previous experience working with the Celonis EMS and analysing processes. The case study workshops were each conducted by two authors and averaged 75 minutes in length. All workshops adhered to a structured framework with the following steps, resembling a BPI project:

(1) Introduction to the *FLAC* Method: We commenced each workshop by acquainting the experts with *FLAC*, utilising a step-by-step approach with the parallelism pattern as a guiding example. To facilitate this, we employed the rulesets derived from our instantiation. We actively encouraged interviewees to offer general feedback or request clarifications as needed throughout the process.

(2) Evaluation of Understandability and Qualitative Feedback: Next, we invited the experts to evaluate *FLAC*'s *understandability* on a scale of 1 to 10, with 1 representing the lowest rating and 10 indicating the highest and provide qualitative feedback.

(3) Practical Application: In the subsequent step, we accessed a PM environment with the exemplary P2P process. After introducing the P2P process, the experts were instructed to explore the process and suggest BPI options following the *parallelism* and *activity automation* BPI patterns for 10 minutes.

(4) Prototype Demonstration: We then demonstrated the prototype's functionality on the P2P process and generated BPI options. We then sought qualitative and quantitative feedback from the experts, assessing the quality of the prototype output (scale of 1 to 10) and comparing the effort of employing the prototype vs. manually generating improvement recommendations on a 5-point Likert scale.

(5) Assessment and Rating: Lastly, we asked the experts to assess (from 1–10) the method and the instantiation in terms of *perceived usefulness*, *ease of use*, and *real-world fidelity* (Davis, 1989; Sonnenberg and vom Brocke, 2012). We defined each term to ensure a shared understanding.

The experts agree that the *FLAC* method has a high level of *understandability*, with an average rating of 7.9/10 (Table 4). They highlighted the clear outline of the steps, which are easy to remember and provide helpful structure to an otherwise primarily unstructured process (Experts 1, 2, 3, 5, 6, 7). They

acknowledged the method's potential and recognised that it helps determine rules that, in turn, can be used to identify BPI options (Experts 1, 2, 3, 4, 5, 6, 7). Two experts pointed out the potential challenges in formulating a comprehensive list of constraints, particularly considering that constraints often vary from one organisation to another. They recommended the provision of additional support mechanisms for users, such as a framework or a catalogue of guiding questions (Expert 2, 4). While we recognise this as a valuable enhancement to the method, implementing it necessitates additional expert interviews to collect input, which have not been conducted yet.

In the practical application, the experts, on average, formulated three improvement recommendations. In contrast, the prototype provided four improvement suggestions for each of the two BPI patterns. On average, there was one match between the results from the experts and the prototype results. The quality of the results from the prototype was rated with an average value of 8/10. In contrast, the effort of employing the prototype vs. creating manual improvement suggestions was rated as "significantly lower" (5x) and "lower" (2x) on a 5-point Likert scale ranging from "significantly lower" to "significantly higher". The experts recognise that the method is *useful* (average rating of 8.3/10) and confirm a high *fidelity with a real-world phenomenon* (average rating of 7.6/10). Some experts' partially lower ratings for ease of use (average rating of 7.4/10) are attributed to the prototype's primary interface, a rudimentary representation awaiting refinement in an actual application. The experts strongly support the method and its instantiation across all dimensions. Detailed information can be found in Table 4.

Expert	**Effort** Prototype vs. manual BPI	**Quality of prototype results**	**Rated understandability**	**Rated usefulness**	**Rated ease of use**	**Rated fidelity** with real-world phenomenon
1	Significantly lower	8	9	8	9	8
2	Significantly lower	8	8	8	9	10
3	Significantly lower	8	8	9	6	7
4	Significantly lower	8	9	8	6	7
5	Lower	9	8	8	10	8
6	Lower	9	7	9	7	8
7	Significantly lower	6	6	8	5	5
Mean		**8.0**	**7.9**	**8.3**	**7.4**	**7.6**
Median		**8.0**	**8.0**	**8.0**	**7.0**	**8.0**

Table 4. Seven BPM experts surveyed evaluation ratings for the FLAC method.

We also invited the experts to share qualitative comments and suggest improvements and further development. A summary of their statements and recommendations can be found in Table 5. The experts confirm that the approach assists users in converting BPI patterns into computable rulesets. The evaluation revealed that the prototype streamlines user understanding by offering specific improvement suggestions by focusing on a reduced set of process steps. This is especially valuable in navigating the complexities of real-world event logs characterised by a substantial volume of cases. The experts emphasised the potential of our method and prototype to significantly reduce the time-to-insight, positioning them as valuable tools for decision support. Despite praising the method's simplicity and supporting its core approach, experts suggested incorporating the process model for context and not creating all the rules process-agnostic. They highlighted the importance of considering input from process experts to customise generated rules for specific organisation contexts, enhancing the overall value of the method.

Positive remarks	Suggestions for improvement
The four steps cover all relevant aspects and are well-chosen.	The generated rules might not be directly applicable to a specific context, and expert input is required.
The method helps reduce the information overload of complex process models.	For easier reference, the process model should be provided as context along with the recommendations.
The structure of the method is easy to understand, and the performed steps are very intuitive.	The rules might be hidden in the minds of people and can be challenging to uncover.
"I think this would be a very valuable add-in for any Process Mining software" (Expert #7).	The interface is sufficient for a prototype but needs to be improved for use in practice.
"The method significantly reduces the time-to-insight as it decreases the number of cases and process steps to look at in the first place and thereby cuts down the complexity" (Expert #6).	Consider distinguishing between "soft" and "hard" constraints, where soft constraints are resolvable issues like resource constraints. This helps avoid filtering out valuable improvement options.
"If at the end several recommendations are generated by the method, and an expert looks over them again, I think that can bring real added value" (Expert #4).	The method focuses solely on individual tasks and steps, potentially overlooking the improvement of entire process fragments.

Table 5. Qualitative comments regarding the FLAC method (partially summarised).

6 Discussion and Conclusion

This work aims at answering the research question, how BPI patterns can be transformed into programmed rulesets that might facilitate the automated development of redesign options in a BPI project. To address this, we present the *FLAC* method, a structured and data-facilitated approach that assists researchers and practitioners in systematically identifying BPI options. Combining DSR as a research paradigm with SME as a research method, our method facilitates the conversion of BPI patterns into rulesets, which, from the content perspective, summarise the knowledge of the BPI pattern while, from the format perspective, are formulated in code. The *FLAC* method itself (Section 4.2) fulfils DO.2 by transforming the knowledge accumulated in BPI patterns into programmable and programmed rulesets that are directly applicable (can be used in BPI) and flexible (can be adapted to specificities of different organisations or processes). At the same time, enacting *FLAC* in the context of a BPI project (Section 4.3) fulfils DO.1 by automatically creating BPI options, thus decreasing the amount of time needed to develop the initial set of BPI options and reducing the amount of expertise required from the project team, and DO.3 by developing BPI options based on the available process and context data.

The *FLAC* method comprehensively addresses the attributes of (1) goal orientation, (2) systematic approach, (3) principles orientation, and (4) repeatability (Braun et al., 2005). Regarding (1) goal orientation, our method focuses on transforming the accumulated knowledge and experience from BPI patterns into programmable rulesets, facilitating the development of BPI options. To achieve this, our artefact assembles and orchestrates artefact parts based on justificatory knowledge into four *FLAC* steps that constitute the method's core. In line with the (2) systematic approach, each activity of the *FLAC* method describes techniques, tools, roles, and defined outputs (Denner et al., 2018). As for (3) principles orientation, our method is geared towards three DOs derived from the literature on BPM and optimised in the feedback round (Section 4.1). Using the SME method composition mode, i.e., reassembling elements of existing methods in line with situational needs (Section 3), and providing a clear description of method activities, techniques, and roles, we achieve (4) repeatability of the *FLAC* method in various contextual settings by design. The *FLAC* method pioneers the translation of BPI patterns into programmable rulesets and, hence, contributes to the prescriptive body of knowledge related to BPI and aims to take a step toward full automation of BPI.

We evaluated and refined the method with the help of five case study workshops and provided an instantiation for demonstration purposes with several BPI patterns in a software prototype. The experts validated the relevance of our research. They positioned the *FLAC* method and its instantiation as a valuable decision-support tool that can help to reduce the time-to-insight when applied at a scale. While

the validation through workshops and instantiation is a commendable starting point, ongoing evaluation in diverse settings and against a broader range of BPI patterns will contribute to a more comprehensive understanding of *FLAC*'s capabilities and potential enhancements.

Building on the evaluation results, our future efforts aim to enhance the method's applicability, moving towards increased automation of process improvement. We plan to expand the application of the method to a broader range of BPI patterns, collaboratively building a repository of programmed rulesets with practitioners and researchers. Additionally, we aspire to explore integration possibilities into PM software to support the seamless transition from insights to actual process improvement.

References

Beerepoot, I., C. Di Ciccio, H. A. Reijers, S. Rinderle-Ma, W. Bandara, A. Burattin, D. Calvanese, T. Chen, I. Cohen, B. Depaire, G. Di Federico, M. Dumas, C. van Dun, T. Fehrer, D. A. Fischer, A. Gal, M. Indulska, V. Isahagian, C. Klinkmüller, W. Kratsch, H. Leopold, A. van Looy, H. Lopez, S. Lukumbuzya, J. Mendling, L. Meyers, L. Moder, M. Montali, V. Muthusamy, M. Reichert, Y. Rizk, M. Rosemann, M. Röglinger, S. Sadiq, R. Seiger, T. Slaats, M. Simkus, I. A. Someh, B. Weber, I. Weber, M. Weske and F. Zerbato (2023). "The biggest business process management problems to solve before we die" *Computers in Industry* 146, 103837.

Braun, C., F. Wortmann, M. Hafner and R. Winter (2005). "Method construction - a core approach to organizational engineering". In: *SAC '05: Proceedings of the 2005 ACM symposium on Applied computing*. Ed. by H. M. Haddad. New York, NY: ACM, pp. 1295–1299.

Bucher, T., M. Klesse, S. Kurpjuweit and R. Winter (2007). "Situational Method Engineering. On the Differentiation of "Context" and "Project Type"". In: *Situational Method Engineering: Fundamentals and Experiences. Proceedings of the IFIP WG 8.1 Working Conference, September 12-14, 2007, Geneva, Switzerland*. Ed. by J. Ralyté, S. Brinkkemper, B. Henderson-Sellers. New York: Springer, pp. 33–48.

Buzacott, J. A. (1996). "Commonalities in Reengineered Business Processes: Models and Issues" *Management Science* 42 (5), 768–782.

Cho, M., M. Song, M. Comuzzi and S. Yoo (2017). "Evaluating the effect of best practices for business process redesign: An evidence-based approach based on process mining techniques" *Decision Support Systems* 104, 92–103.

Davis, F. D. (1989). "Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology" *MIS Quarterly* 13 (3), 319.

Denner, M.-S., L. C. Püschel and M. Röglinger (2018). "How to Exploit the Digitalization Potential of Business Processes" *Business & Information Systems Engineering* 60 (4), 331–349.

Dumas, M., M. La Rosa, J. Mendling and H. A. Reijers (2018). *Fundamentals of Business Process Management.* 2nd Edition. Berlin, Heidelberg: Springer.

Fehrer, T. (2023). "Process-pattern.app - A collection of business process redesign patterns". In: *BPM-D 2023*. Ed. by Dirk Fahland, Andrés Jiménez-Ram-Ma, Tijs Slaats, Johan Versendaal, Barbara Weber, Mathias Weske, Karolin Winter: CEUR-WS.org, pp. 117–121.

Fehrer, T., D. A. Fischer, S. J. Leemans, M. Röglinger and M. T. Wynn (2022). "An assisted approach to business process redesign" *Decision Support Systems* 156, 113749.

Gregor, S. and A. R. Hevner (2013). "Positioning and Presenting Design Science Research for Maximum Impact" *MIS Quarterly* 37 (2), 337–355.

Gross, S., M. Malinova and J. Mendling (2019). "Navigating Through the Maze of Business Process Change Methods". In: *Proceedings of the 52nd Hawaii International Conference on System Sciences*. Ed. by T. Bui: Hawaii International Conference on System Sciences, pp. 6270–6279.

Hevner, A. R., S. T. March, J. Park and S. Ram (2004). "Design Science in Information Systems Research" *MIS Quarterly* 28 (1), 75–105.

Jansen-Vullers, M. H. and H. A. Reijers (2005). "Business Process Redesign in Healthcare: Towards a Structured Approach" *INFOR: Information Systems and Operational Research* 43 (4), 321–339.

Limam Mansar, S. and H. A. Reijers (2007). "Best practices in business process redesign: use and impact" *Business Process Management Journal* 13 (2), 193–213.

Netjes, M., S. Limam Mansar, H. A. Reijers and W. M. P. van der Aalst (2008). "Performing Business Process Redesign with Best Practices: An Evolutionary Approach". In: *Enterprise Information Systems. 9th International Conference, ICEIS 2007, Funchal, Madeira, June 12-16, 2007, Revised Selected Papers*. Ed. by J. Filipe, J. Cordeiro, J. Cardoso. Berlin, Heidelberg: Springer, pp. 199–211.

Netjes, M., R. S. Mans, H. A. Reijers, W. M. P. van der Aalst and R. J. B. Vanwersch (2010). "BPR Best Practices for the Healthcare Domain". In: *BPM 2009 Workshops*. Ed. by S. Rinderle-Ma, S. Sadiq, F. Leymann. Berlin, Heidelberg: Springer, pp. 605–616.

Niedermann, F., S. Radeschutz and B. Mitschang (2010). "Design-Time Process Optimization through Optimization Patterns and Process Model Matching". In: *2010 IEEE 12th Conference on Commerce and Enterprise Computing (CEC 2010)*: IEEE, pp. 48–55.

Nissen, M. E. (1996). *Knowledge-based organizational process redesign: Using process flow measures to transform procurement*.

Park, G. and W. M. P. van der Aalst (2022). "Action-oriented process mining: bridging the gap between insights and actions" *Progress in Artificial Intelligence*.

Peffers, K., T. Tuunanen, M. A. Rothenberger and S. Chatterjee (2007). "A Design Science Research Methodology for Information Systems Research" *J Manag Inf Syst* 24 (3), 45–77.

Reijers, H. A. and S. Limam Mansar (2005). "Best practices in business process redesign: an overview and qualitative evaluation of successful redesign heuristics" *Omega* 33 (4), 283–306.

Rosemann, M. (2014). "Proposals for Future BPM Research Directions". In C. Ouyang and J.-Y. Jung (eds.) *Asia Pacific Business Process Management. Second Asia Pacific Conference, AP-BPM 2014, Brisbane, QLD, Australia, July 3-4, 2014, Proceedings*, pp. 1–15. Cham: Springer.

Sonnenberg, C. and J. vom Brocke (2012). "Evaluations in the Science of the Artificial – Reconsidering the Build-Evaluate Pattern in Design Science Research". In: *DESRIST 2012*. Ed. by K. Peffers, M. Rothenberger, B. Kuechler. Berlin, Heidelberg: Springer, pp. 381–397.

Stein Dani, V., H. Leopold, J. M. E. M. van der Werf and H. A. Reijers (2024). "Progressing from Process Mining Insights to Process Improvement: Challenges and Recommendations". In: *Enterprise Design, Operations, and Computing. 27th International Conference, EDOC 2023, Groningen, The Netherlands, October 30 - November 3, 2023, Proceedings*. Ed. by H. A. Proper, L. Pufahl, D. Karastoyanova, M. van Sinderen, J. Moreira, pp. 152–168.

van der Aalst, W. and K. M. van Hee (2002). *Workflow management. Models, methods and systems.* 1. MIT Press paperback ed. Cambridge, Mass., London: MIT Press.

van der Waal, W., I. Beerepoot, I. van de Weerd and H. A. Reijers (2022). "The SWORD is Mightier Than the Interview: A Framework for Semi-automatic WORkaround Detection". In: *BPM 2022*. Ed. by C. Di Ciccio, R. Dijkman, A. del Río Ortega, S. Rinderle-Ma, pp. 91–106.

Vanwersch, R. J. B., K. Shahzad, I. Vanderfeesten, K. Vanhaecht, P. Grefen, L. Pintelon, J. Mendling, G. G. van Merode and H. A. Reijers (2016). "A Critical Evaluation and Framework of Business Process Improvement Methods" *Business & Information Systems Engineering* 58 (1), 43–53.

Venable, J., J. Pries-Heje and R. Baskerville (2012). "A Comprehensive Framework for Evaluation in Design Science Research". In: *DESRIST 2012*. Ed. by K. Peffers, M. Rothenberger, B. Kuechler. Berlin, Heidelberg: Springer, pp. 423–438.

vom Brocke, J., M.-S. Baier, T. Schmiedel, K. Stelzl, M. Röglinger and C. Wehking (2021a). "Context-Aware Business Process Management. Method Assessment and Selection" *Business & Information Systems Engineering* 63 (5), 533–550.

vom Brocke, J., W. M. P. van der Aalst, T. Grisold, W. Kremser, J. Mendling, B. Pentland, J. Recker, M. Röglinger, M. Rosemann and B. Weber (2021b). "Process Science: The Interdisciplinary Study of Continuous Change" *SSRN Electronic Journal*.

Zellner, G. (2013). "Towards a framework for identifying business process redesign patterns" *Business Process Management Journal* 19 (4), 600–623.

Zuhaira, B. and N. Ahmad (2021). "Business process modeling, implementation, analysis, and management: the case of business process management tools" *Business Process Management Journal* 27 (1), 145–183.